Mach Learn (2008) 73: 87–106
DOI 10.1007/s10994-008-5066-6

A bias/variance decomposition for models using collective inference

Jennifer Neville · David Jensen

Received: 15 September 2007 / Revised: 9 April 2008 / Accepted: 16 April 2008 /
Published online: 16 July 2008

Abstract Bias/variance analysis is a useful tool for investigating the performance of machine learning algorithms. Conventional analysis decomposes loss into errors due to aspects of the learning process, but in relational domains, the inference process used for prediction introduces an additional source of error. *Collective inference* techniques introduce additional error, both through the use of approximate inference algorithms and through variation in the availability of test-set information. To date, the impact of *inference* error on model performance has not been investigated. We propose a new bias/variance framework that decomposes loss into errors due to both the *learning* and *inference* processes. We evaluate the performance of three relational models on both synthetic and real-world datasets and show that (1) inference can be a significant source of error, and (2) the models exhibit different types of errors as data characteristics are varied.

Keywords Statistical relational learning · Collective inference · Evaluation

1 Introduction

Bias/variance analysis (Geman et al. 1992) has been used for a number of years to investigate the mechanisms behind model performance. This analysis has focused on loss as a measure of classification performance—a loss function $L(t, y)$ defines the penalty for predicting class y for an instance x when its true class label value is t. Bias/variance analysis decomposes the expected loss ($E[L(t, y)]$) for an instance x into three components of error: bias, variance, and noise. Overall model loss can be reduced by reducing either bias

Editors: Hendrik Blockeel, Jude Shavlik, Prasad Tadepalli.

J. Neville (✉)
Departments of Computer Science and Statistics, Purdue University, West Lafayette, IN 47907-2107, USA
e-mail: neville@cs.purdue.edu

D. Jensen
Department of Computer Science, University of Massachusetts Amherst, Amherst, MA 01003-4610, USA

or variance, but there is often a tradeoff between the two components when learning statistical models. Searching over a larger model space, to estimate a more complex model, can decrease bias but often increases variance. On the other hand, very simple models can sometimes outperform complex models due to decreased variance, albeit with higher bias (Holte 1993).

Conventional bias/variance analysis accounts for two sources of variability in expected loss—the variation of true values t due to noise in the domain and the variation in predicted values y due to learning the model on different training sets (i.e., different samples from the underlying distribution). Correspondingly, in the traditional decomposition, bias and variance measure estimation errors associated with the learning technique. For example, the Naive Bayes classifier typically has high bias due to the assumption of independence among features, but low variance due to the use of a large sample (i.e., entire training set) to estimate the conditional probability distribution for each feature (Domingos and Pazzani 1997).

The assumption underlying the conventional decomposition is that there is no variation in expected loss due to (1) the inference process used for prediction, and (2) the available information in the test set. Classification of relational data often violates these assumptions when *collective inference* models are used.

Probabilistic models for independent and identically distributed (i.i.d.) data estimate a conditional distribution for the target class label Y, given other attributes $\mathbf{X}$, focusing on a single instance i:[1]

$$p(y^i|\mathbf{x}^i) = p(y^i|x_1^i, x_2^i, \ldots, x_m^i).$$

In collective inference models (see e.g., Jensen et al. 2004), the predictive distribution for Y is also conditioned on the attributes and the class labels of related instances $R = \{j : 1 \leq j \leq n \wedge R(i, j)\}$:

$$p(y^i|\mathbf{x}^i, \mathbf{x}^R, \mathbf{y}^R) = p(y^i|x_1^i, \ldots, x_m^i, x_1^{j_1}, \ldots, x_m^{j_1}, \ldots, x_1^{j_r}, \ldots, x_m^{j_r}, y^{j_1}, \ldots, y^{j_r}).$$

When inferring the values of y^i for a number of interrelated instances, some of the values of $\mathbf{y}^R$ may be unknown (if the class labels of related instances are to be inferred as well). Collective inference techniques infer the unobserved class labels values simultaneously, with the aim of fully exploiting the dependencies among related instances.

However, the use of collective inference introduces a new source of variability that can affect expected loss. Collective inference often requires the use of approximate inference techniques, which may result in variation in the predicted values y for an instance x. For example, the final prediction for x may depend on the initial (random) start state used during inference, thus multiple runs of inference with the same model could result in different predictions y. In addition, relational models are often applied to classify a partially labeled test set, where the known class labels serve to seed the collective inference process. In this case, the predicted values y may vary based on *which* instances are labeled in the test set. Due to the heterogeneity of relational data graphs, it is likely that some instances will have more of an impact on neighbor predictions than others, thus the set of instances that seed the inference process may be a source of substantial variation. Note that this variation does not occur in predictions from i.i.d. models—since the predictions are computed independently for each instance, they do not vary based on the observed and/or predicted information for other instances in the data.

[1] Here we use superscripts to refer to instances and subscripts to refer to attributes.

To date, the impact of *inference* variation on model performance has not been investigated. We propose a new bias/variance framework that decomposes expected loss into components of error due to both the *learning* algorithm used to estimate the model and the *inference* algorithm used for prediction. We evaluate the performance of three relational models on synthetic data and of two models on real-world data, using the framework to understand the reasons for poor model performance. Each of the models exhibits a different relationship between error and dataset characteristics—relational Markov networks (Taskar et al. 2002) have higher inference bias in densely connected networks; relational dependency networks (Neville and Jensen 2004) have higher inference variance when there is little information to seed the inference process; latent group models (Neville and Jensen 2005) have higher learning bias when the underlying group structure is difficult to identify from the network structure. Using this understanding, we propose a number of algorithmic modifications to improve the performance of each model.

2 Framework

In conventional bias/variance analysis, loss is decomposed into three factors: bias, variance, and noise (Geman et al. 1992; Friedman 1997; Domingos 2000; James 2003). Given an example x, a model that produces a prediction $f(x) = y$, and a true value for x of t, squared loss is defined[2] as: $L(t, y) = (t - y)^2$. The expected loss for an example x can be decomposed into bias, variance, and noise components. Here the expectation is over training sets D—the expected loss is measured with respect to the variation in predictions for x when the model is learned on different training sets: $E_{D,t}[L(t, y)] = B(x) + V(x) + N(x)$.

Bias is defined as the loss incurred by the mean prediction y_m relative to the Bayes-optimal prediction y_* (see e.g., Duda et al. 2001): $B(x) = L(y_*, y_m)$. Variance is defined as the average loss incurred by all predictions y, relative to the mean prediction y_m: $V(x) = E_D[L(y_m, y)]$. Noise is defined as the loss that is incurred independent of the learning algorithm, due to noise in the data set: $N(x) = E_t[L(t, y_*)]$.

Bias and variance measures are typically estimated for each test-set example x using models learned from a number of different training sets. This type of analysis decomposes model error to associate it with aspects of the *learning* process, not aspects of the *inference* process used for prediction. The technique assumes that there is no variation in prediction if the same model is applied multiple times to the same dataset. However, in relational datasets there can be additional variation in model predictions due to the use of collective inference techniques and due to the availability of test-set information. In order to accurately ascribe errors to learning *and* inference, we have extended the conventional bias/variance framework to incorporate errors due to application of the model for inference.

For relational data, we first define *total* loss, bias, and variance with respect to variation due to both the learning and inference processes. Note that we focus on marginal squared loss in this framework.

[2]Note that we use y as a general reference to the model prediction for instance x. However, when analyzing probabilistic models we use the probability estimate for y rather than the most likely class value (i.e., $y = p(y = +)$ and not $y = \arg\max_y p(y)$). To compute loss (with binary class values), we set $t = 1$ if the true class label is $+$ and 0 otherwise.

Definition 1 *Total loss* for an instance $x \in D_I$ is defined as $E_{LI}[L(t, y)]$. This is the expected loss for x over the training sets D_L used for learning, the inference runs I in the test set D_I, and the true value t for instance x.[3]

Definition 2 *Total bias* for an instance $x \in D_I$ is defined as $B_T(x) = (E_{LI}[t] - E_{LI}[y])^2$. This is the loss incurred by this mean prediction for x relative to its Bayes-optimal prediction ($y_* = E_L[t] = E_{LI}[t]$), where the mean prediction $E_{LI}[y]$ is averaged over both the training sets D_L used for learning and inference runs I in the test set D_I.

Definition 3 *Total variance* for an instance $x \in D_I$ is defined as $V_T(x) = E_{LI}[(E_{LI}[y] - y)^2]$. This is the average loss incurred by all predictions y, relative to the mean prediction $E_{LI}[y]$ over both learning and inference.

Notice the difference between our approach and conventional bias/variance decompositions is with respect to the expectations. In the conventional setting the expectation is over learning alone (i.e., variation due to learning the model on different training sets). We are now using an expectation over both learning and inference, capturing the variation in predictions due to different inference runs with the same learned model.

Following the standard decomposition for loss as described in (Geman et al. 1992), we can decompose *total* loss into *total* bias, variance, and noise.

Lemma 1 $E_{LI}[L(t, y)] = B_T(x) + V_T(x) + N(x)$.

Proof

$$
\begin{aligned}
E_{LI}[L(t, y)] &= E_{LI}[(t - y)^2] \\
&= E_{LI}[t^2 - 2ty + y^2] \\
&= E_{LI}[y^2] - 2E_{LI}[t]E_{LI}[y] + E_{LI}[t^2] \\
&= E_{LI}[y^2] - E_{LI}[y]^2 + E_{LI}[y]^2 - 2E_{LI}[t]E_{LI}[y] + E_{LI}[t^2] \\
&= V_T(x) + E_{LI}[y]^2 - 2E_{LI}[t]E_{LI}[y] + E_{LI}[t^2] \\
&= V_T(x) + E_{LI}[y]^2 - 2E_{LI}[y]E_{LI}[t] + E_{LI}[t]^2 - E_{LI}[t]^2 + E_{LI}[t^2] \\
&= V_T(x) + (E_{LI}[t] - E_{LI}[y])^2 - E_{LI}[t]^2 + E_{LI}[t^2] \\
&= V_T(x) + B_T(x) + E_{LI}[t^2] - E_{LI}[t]^2 \\
&= V_T(x) + B_T(x) + E_{LI}[(t - E_{LI}[t])^2] \\
&= B_T(x) + V_T(x) + N(x).
\end{aligned}
$$

□

In this decomposition the total bias $B_T(x)$ and total variance $V_T(x)$ are calculated with respect to variation in model predictions due to both the learning and inference processes. Note that the definition of noise is the same as in previous bias/variance decompositions—it is the loss incurred independent of the chosen modeling technique, due to noise in the data set.

[3]For notational simplicity, we refer to the expectation $E_{D_L,I,t}[.]$, over datasets D_L, inference runs I, and true value t as $E_{LI}[.]$.

Now we define the *learning* bias and variance through expectations over training sets alone (D_L), using Bayes-optimal predictions for related instances in the test set during inference. This enables the application of exact inference techniques for prediction (since we no longer need to perform collective classification) and ensures that the test-set information most closely matches the information used during learning. Note that this part of the analysis mirrors the conventional approach to bias/variance decomposition, isolating the errors due to the learning process.

Definition 4 *Learning bias* for an instance $x \in D_I$ is defined as $B_L(x) = (E_L[t] - E_L[y])^2$. This is the loss incurred by the mean prediction $E_L[y]$ averaged over the training sets D_L used for learning, relative to the Bayes-optimal prediction for that instance $E_L[t]$. During inference we allow the model to use the Bayes-optimal predictions for all other instances in the dataset ($X - \{x\}$), which isolates the error due to the learning process.

Definition 5 *Learning variance* for an instance $x \in D_I$ is defined as $V_L(x) = E_L[(E_L[y] - y)^2]$. This is the average loss incurred by all predictions y, relative to the mean prediction $E_L[y]$ over learning. Again, we use the Bayes-optimal predictions for all other instances in the dataset ($X - \{x\}$) during inference.

Now we can define the *inference* bias and variance with respect to the learning components.

Definition 6 *Inference bias* for an instance $x \in D_I$ is defined as $B_I(x) = (E_L[y] - E_{LI}[y])^2$. This is the loss incurred by the mean prediction $E_{LI}[y]$ (averaged over learning and inference), relative to the mean prediction over learning alone $E_L[y]$.

Definition 7 *Inference variance* for an instance $x \in D_I$ is defined as $V_I(x) = \alpha - \beta = E_{LI}[(E_L[y] - y)^2] - E_L[(E_{LI}[y] - y)^2]$. This includes two variance components. The first component α is the average loss incurred by all predictions y, relative to the mean learning prediction $E_L[y]$. The second component β 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 is the difference between the α and β components.

We can now show that total bias is composed of the learning and inference bias components.

Lemma 2 *$B_T(x) = B_L(x) + B_I(x) + \gamma$, where γ is an interaction bias term defined as $\gamma = 2[(E_L[t] - E_L[y])(E_L[y] - E_{LI}[y])]$.*

Proof

$$
\begin{aligned}
B_T(x) &= (E_{LI}[t] - E_{LI}[y])^2 \\
&= E_L[t]^2 - 2E_L[t]E_{LI}[y] + E_{LI}[y]^2 + 2E_L[y]E_L[t] \\
&\quad - 2E_L[y]E_L[t] + E_L[y]^2 - E_L[y]^2 \\
&= (E_L[t] - E_L[y])^2 - 2E_L[t]E_{LI}[y] + E_{LI}[y]^2 - E_L[y]^2 + 2E_L[y]E_L[t] \\
&= B_L(x) + (E_L[y] - E_{LI}[y])^2 - 2E_L[t]E_{LI}[y] - 2E_L[y]^2 \\
&\quad + 2E_L[y]E_L[t] + 2E_L[y]E_{LI}[y]
\end{aligned}
$$

$$= B_L(x) + B_I(x) + 2[(E_L[t] - E_L[y])(E_L[y] - E_{LI}[y])]$$
$$= B_L(x) + B_I(x) + \gamma. \qquad \square$$

Note that all bias terms are squared deviations. The relationship among the unsquared bias components is trivial: $E_{LI}[t] - E_{LI}[y] = (E_L[t] - E_L[y]) + (E_L[y] - E_{LI}[y])$. The γ component is thus an interaction term due to the quadratic expansion of B_T. Positive values of γ occur when the mean prediction of the total distribution ($E_{LI}[y]$) is farther from the optimal prediction ($E_L[t]$) than the mean prediction of the learning distribution ($E_L[y]$). Negative values of γ indicate that either (1) $E_{LI}[y]$ and $E_L[y]$ are on opposite sides of $E_L[t]$, or (2) $E_L[y]$ is farther from $E_L[t]$ than $E_{LI}[y]$. We will revisit this issue below and illustrate with an example.

Next we show that total variance is composed of the learning and inference variance components.

Lemma 3 $V_T(x) = V_L(x) + V_I(x)$.

Proof

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

Now we can show that total loss decomposes into learning bias/variance, inference bias/variance, and noise.

Theorem 1 *In collective classification settings, where there is variation in model predictions due to both the learning and inference processes, total loss for an instance x can be decomposed into the following components: overall noise, learning bias, inference bias, learning variance, inference variance, and an interaction bias term γ:*

$$E_{LI}[L(t, y)] = N(x) + B_L(x) + B_I(x) + V_L(x) + V_I(x) + \gamma.$$

Proof This follows directly from Lemmas 1–3. $\square$

To illustrate the decomposition, consider the distributions of model predictions in Fig. 1. We measure the variation of model predictions for an instance x in two ways. First, when we

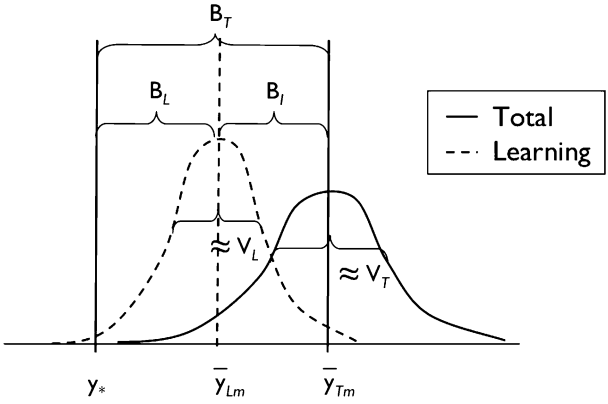


Fig. 1 Distributions of model predictions illustrating bias and variance components. The Bayes-optimal prediction is denoted y_*; the *dashed line* denotes the *learning* distribution (using optimal neighbor predictions for exact inference); the *solid line* denotes the *total* distribution (using collective inference)

generate synthetic data we can record the data generation probability as the optimal prediction y_*. Next, we record marginal predictions for x from models learned on different training sets, allowing the optimal predictions of related instances (y_*) to be used during exact inference. These predictions form the *learning* distribution, with a mean *learning* prediction of y_{Lm}. Finally, we record predictions for x from models learned on different training sets, where each learned model is applied a number of times on the test set using collective inference. These predictions form the *total* distribution, with a mean *total* prediction of y_{Tm}. The model's *learning* bias is calculated from the difference between y_* and y_{Lm}; the *inference* bias is calculated from the difference between y_{Lm} and y_{Tm}. The model's *learning* variance is calculated from the dispersion of the *learning* distribution; the *inference* variance is calculated as the difference between the *total* variance and the *learning* variance. Note that the inference variance can be negative if the total variance is less than the learning variance. This could happen if using the true class labels of neighbors for prediction results in a more stable prediction than using the optimal probabilities.

Also, note that if the bias terms were not squared, the inference bias could be negative. This would happen if the application of collective inference results in improved model predictions (compared to exact inference with optimal neighbor probabilities). This is reflected in the γ term in the decomposition. If $y_* \leq y_{Lm} \leq y_{Tm}$ or $y_{Tm} \leq y_{Lm} \leq y_*$, then γ will be positive. This is the case that we would expect to see in practice—reflecting a degradation in the quality of the predictions due to the use approximate collective inference instead of exact inference with optimal neighbor information. However, if $y_* \leq y_{Tm} \leq y_{Lm}$ or $y_{Lm} \leq y_{Tm} \leq y_*$, then collective inference has improved performance and γ will be negative. The value of γ will also be negative if $y_{Tm} \leq y_* \leq y_{Lm}$ or $y_{Lm} \leq y_* \leq y_{Tm}$. This indicates that different types of errors are made with collective inference and exact inference (i.e., one overestimates the probability while the other underestimates it).

Finally, we note that since learning bias and variance are defined with respect to model performance when all neighbors are optimally labeled, a model can exhibit low learning bias without necessarily increasing the expressiveness of the hypothesis space it explores. This can be achieved through the use of strong prior knowledge to restrict the hypothesis space to very specific relational dependencies. For example, when the data exhibit strong correlation among the class labels of related instances, very simple models that represent *autocorrelation*[4] dependencies can achieve low learning bias. Indeed, there are very simple

[4]See Sect. 3.1.1 for a more detailed description of autocorrelation.

relational models that do not learn at all but simply assume autocorrelation is present while propagating information throughout the graph for inference (e.g., Macskassy and Provost 2007). Although these types of simple models will have low learning bias when these relational dependencies are present, the models will not enjoy uniformly superior classification performance—they will only be successful in inference settings where there is enough information (i.e., known class labels) to seed the inference process. In this case, the low learning bias accurately reflects the ability of the models to represent the dependencies in the data. When there are few labeled instances in the test set, model error will be due to inference bias or inference variance (depending of the chosen inference technique). This more accurately describes the source of the error in the model, which is due to characteristics of the test set and the inference process.

3 Experiments

The experiments below illustrate the utility of our bias/variance framework for relational model evaluation. We compare three models on synthetic data and two models on real-world data, measuring squared loss and decomposing it into bias and variance components for each model. The experiments assess model performance in a collective classification context, where a single attribute is unobserved in the test set.

3.1 Data

3.1.1 Synthetic data

The synthetic datasets are homogeneous data graphs with *relational autocorrelation*. Relational autocorrelation is a statistical dependency between the values of the same variable (e.g., the class label) on related entities and is a nearly ubiquitous characteristic of relational datasets (Jensen and Neville 2002; Taskar et al. 2002; Bernstein et al. 2003; Hill et al. 2006). For example, hyperlinked web pages are more likely to share the same topic than randomly selected pages.

Much of the success of collective inference techniques in relational domains is due to the presence of autocorrelation in the data. When there are dependencies among the class labels of related instances, the inferences about one instance can be used to improve the inferences about other related instances. Collective inference techniques exploit these dependencies, producing more accurate predictions than conditional inference for each instance independently (Jensen et al. 2004).

More formally, we define relational autocorrelation with respect to an attributed graph $G = (V, E)$, where each node $v \in V$ represents an object (i.e., instance) and each edge $e \in E$ represents a binary link (i.e., relation). Autocorrelation is measured for a set of object pairs P_R related through paths of length l in a set of edges E_R: $P_R = \{(v_i, v_j) : e_{ik_1}, e_{k_1k_2}, \ldots, e_{k_lj} \in E_R\}$, where $E_R = \{e_{ij}\} \subseteq E$. It is the correlation between the values of a variable X on the object pairs $(v_i.x, v_j.x)$ such that $(v_i, v_j) \in P_R$.

The autocorrelation in our synthetic datasets is due to an underlying (hidden) group structure—where each group's members are likely to have the same class label and group members are more likely to link to each other than to objects in other groups. More specifically, each object has a group G and four boolean attributes: a class label Y, and attributes X_1, X_2 and X_3. Each group has an associated type T and each object's Y value is determined

from the type of its group. This procedure results in data where Y has an autocorrelation level of 0.5 among pairs of directly linked objects.[5]

The generative process uses a simple model where Y depends only on the type of the associated group, X_1 depends on Y, and the other two attributes have no dependencies. We used the procedure below to generate a dataset with N_O objects and G_S average group size:

1. For each group g, $1 \leq g \leq (N_G = N_O/G_S)$:
   (a) Choose a value for group type t_g from $p(T)$.
2. For each object i, $1 \leq i \leq N_O$:
   (a) Choose a group g_i uniformly in $[1, N_G]$.
   (b) Choose a class value Y_i from $p(Y|T_{G_i})$.
   (c) Choose a value for X_{1i} from $p(X_1|Y)$.
   (d) Choose values for X_{2i} from $p(X_2)$ and X_{3i} from $p(X_3)$.
3. For each object j, $1 \leq j \leq N_O$:
   (a) For each object k, $j < k \leq N_O$:
      (i) Choose whether the two objects are linked from $p(E|G_j = G_k)$.

The following default parameter settings were used in the generation procedure:

$$
\begin{aligned}
N_O &= 250, \\
p(T) &= \{p(T=1) = 0.50;\ p(T=0) = 0.50\}, \\
p(Y|T_G) &= \{p(Y=1|T_G=1) = 0.90;\ p(Y=0|T_G=0) = 0.90\}, \\
p(X_1|Y) &= \{p(X_1=1|Y=1) = 0.75;\ p(X_1=0|Y=0) = 0.75\}, \\
p(X_2=1) &= p(X_3=1) = 0.50.
\end{aligned}
$$

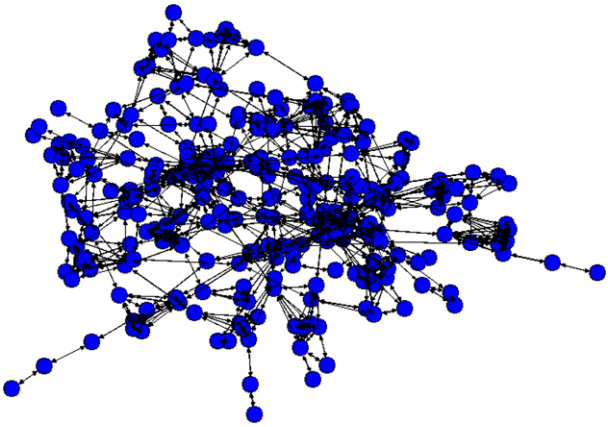
For the experiments, we generated four types of datasets with two groups sizes and two levels of linkage:

$$
\begin{aligned}
G_S: &\quad small = 5; \qquad large = 25 \\
L_{low}|G_S = small: &\quad \{p(E=1|G_j = G_k) = 0.50;\ p(E=1|G_j \neq G_k) = 0.0008\}, \\
L_{high}|G_S = small: &\quad \{p(E=1|G_j = G_k) = 0.80;\ p(E=1|G_j \neq G_k) = 0.004\}, \\
L_{low}|G_S = large: &\quad \{p(E=1|G_j = G_k) = 0.20;\ p(E=1|G_j \neq G_k) = 0.0008\}, \\
L_{high}|G_S = large: &\quad \{p(E=1|G_j = G_k) = 0.30;\ p(E=1|G_j \neq G_k) = 0.004\}.
\end{aligned}
$$

Figure 2 graphs a sample synthetic dataset with small group size and high linkage. The final datasets are homogeneous—there is only one object type and one link type, and each object has four attributes. After the groups are used to generate the data, we delete them from the data—the groups are not available for model learning or inference.

[5] We only report results for autocorrelation = 0.5 because varying autocorrelation does not alter the relative performance of the models—lower levels of autocorrelation weaken the effects, higher levels strength the effects reported herein.

Fig. 2 Sample synthetic dataset



3.1.2 Real world data

We used two real-world relational datasets for model evaluation. The first data set was collected by the WebKB Project (Craven et al. 1998). The data consist of a set of 3,877 web pages from four computer science departments, manually labeled with the categories: course, faculty, staff, student, research project, or other. We considered the unipartite co-citation web graph. The classification task was to predict page category. As in previous work on this dataset (Taskar et al. 2002), we do not try to predict the category "other"; we remove these instances from the data after creating the co-citation graph.

The second data set is drawn from Cora, a database of computer science research papers extracted automatically from the Web using machine learning techniques (McCallum et al. 1999). We considered the unipartite co-citation graph of 4,330 machine-learning papers. For classification, we sampled the 1669 papers published between 1993 and 1998. The classification task was to predict one of seven paper topics (e.g., neural networks).

3.2 Models

We compare the performance of three different relational models: relational Markov networks (RMNs), relational dependency networks (RDNs), and latent group models (LGMs).

RMNs (Taskar et al. 2002), extend Markov networks to a relational setting, representing a joint distribution over the values of the attributes in a network dataset. RMNs represent the joint distribution using an undirected graphical model with a set of relational clique templates and corresponding potential functions. We defined clique templates for each pairwise combination of class label value and attribute value, where the available attributes consisted of the intrinsic attributes of objects, and both the class label and attributes of directly related objects. We used maximum a posteriori parameter estimation to estimate the feature weights, using conjugate gradient with zero-mean Gaussian priors, and a uniform prior variance of 5. For inference, we used loopy belief propagation (Murphy et al. 1999).

RDNs (Neville and Jensen 2004) extend dependency networks (Heckerman et al. 2000) to work with relational data in much the same way that RMNs extend Markov networks. RDNs approximate the joint distribution with pseudolikelihood—modeling the joint with a set of conditional probability distributions that are each learned independently. We used relational probability trees (RPTs) (Neville et al. 2003) as the component CPD to model Y. Note that the RPT is a selective model (i.e., the learning algorithm selects which features

are relevant to the task), so it may not use all the available attributes. For inference, we used Gibbs sampling with fixed-length chains of 2000 samples and a burn-in length of 100.

LGMs (Neville and Jensen 2005) specify a generative probabilistic model for the attributes and link structure of a relational dataset. LGMs are a form of probabilistic relational model that combine a relational Bayesian network (Getoor et al. 2001), link existence uncertainty, and hierarchical latent variables. The model posits groups of objects in the data of various types. Membership in these groups influences the observed attributes of objects, as well as the existence of relations (links) among objects. LGMs use a sequential learning approach—spectral clustering is used first to determine group membership based on the observed link structure alone, then EM is used to learn the remainder of the model (i.e., infer group types and estimate parameters). The resulting clusters are disjoint, and within each group the class labels are conditionally independent given the group type, thus we can use standard belief propagation for inference in the test set.

The synthetic data generation uses an LGM model with manually specified parameters. However, we note that the LGM models evaluated below are different LGM models that are *learned* from the training data. Due to modest training-set sizes, the learning procedure will not recover the model used for data generation and thus the learned LGMs will have non-zero bias.

3.3 Methodology

The experiments evaluate model performance in a collective classification context, where a single attribute is unobserved in the test set. During inference we varied the number of known class labels in the test set, measuring performance on the remaining unlabeled instances. This serves to illustrate how model performance varies as the amount of information seeding the inference process increases. We expect similar performance when other information seeds the inference process—for example, when some labels can be inferred from intrinsic attributes, or when weak predictions about related instances serve to constrain the system.

3.3.1 Synthetic data

The synthetic data experiments explore the effects of relational graph and attribute structure on model performance. We generated synthetic datasets with varying levels of linkage and group structure. Group structure is used to control the inherent clustering of the data. We generated data in the manner described above, and learned models to predict Y using the intrinsic attributes of the object (X_1, X_2, X_3) as well as the class label and attributes of directly related objects (Y, X_1, X_2, X_3). We generated disjoint training and test sets for use in the procedure below and compared LGM, RDN, and RMN performance.

To decompose the learning and inference errors in the synthetic data experiments, we used the following procedure:

1. For each outer trial $i = \{1, \ldots, 5\}$:
   (a) Generate test set i; record optimal predictions.
   (b) For each learning trial $j = \{1, \ldots, 5\}$:
      (i) Generate training set j.
      (ii) Learn model of Y on training set j.

(iii) Infer marginal probabilities for test set i with optimal labels;[6] record *learning* predictions.
(iv) For each inference trial $k = \{1, \ldots, 5\}$ and proportion labeled $p = \{0.0, 0.3, 0.6\}$:
(A) Choose $p\%$ of test set randomly and reveal the correct class labels for those objects.
(B) Infer marginal probabilities for unlabeled test objects and then record *total* predictions.
(C) Measure squared loss.
(c) Calculate *learning* bias and variance from distributions of *learning* predictions. Calculate *inference* bias and variance from distributions of *total* predictions.
2. Calculate average model loss, average *learning* bias/variance, and average *inference* bias/variance.

3.3.2 Real world data

The real-world data experiments evaluate the models in two illustrative real-world scenarios. In these experiments, we only evaluate LGM and RDN models. This is due to the computational complexity of learning the RMNs on large datasets with many relational attributes. Instead of closed-form parameter estimation, RMNs are trained with conjugate gradient methods, where each iteration requires a round of collective inference. In the scenarios we consider below, the cost of inference is prohibitively expensive to use successively during learning.

The first scenario is classification in the WebKB data. The task was to predict page category in the co-citation graph using the intrinsic attributes of the pages and the class labels and attributes of hyperlinked pages. We sampled by department and treated each of the four departments as a disjoint test set for the *outer trials*, while learning different models on each of the $\binom{3}{2}$ other departments for the *learning trials*.

The second scenario is classification in the Cora data. The task was to predict paper topic in the co-citation graph using the intrinsic attributes of the papers and the class labels and attributes of cited papers. We used temporal sampling in which we learned models on one year of data and applied the models to the subsequent year. We considered each year from 1994–1998 as a test set in the *outer trials*. For each of these years, we used the sample from the previous year as the training set and employed *snowball sampling* (Goodman 1961) to partition the data into three samples for the *learning trials*.

To decompose the learning and inference errors in the real-data experiments, we used the following procedure:

1. For each outer trial i:
(a) Record class labels of test set i as the optimal predictions.
(b) For each learning trial j:
(i) Learn model of Y on training set j.
(ii) Infer marginal probabilities for test set i with optimal labels; record *learning* predictions.
(iii) For each inference trial $k = \{1, \ldots, 5\}$ and proportion labeled $p = \{0.0, 0.3, 0.6, 0.9\}$:

[6] Predictions for instance i will use the optimal predictions for related objects in the graph (i.e., $\mathbf{x} - \{x_i\}$) as the *correct* class labels for those instances.

(A) Choose $p\%$ of test set randomly and reveal the correct class labels for those objects.
(B) Infer marginal probabilities for unlabeled test objects and then record *total* predictions.
(C) Measure squared loss.

(c) Calculate *learning* bias and variance from distributions of *learning* predictions. Calculate *inference* bias and variance from distributions of *total* predictions.

2. Calculate average model loss, average *learning* bias/variance, and average *inference* bias/variance.

In the real data experiments we do not have a generative model for the data so we need to approximate the optimal probabilities for use in the bias/variance decomposition. As in previous bias/variance analysis (see e.g., James 2003), we use empirical estimates of the Bayes-optimal predictions in place of the true optimal predictions. The Bayes-optimal prediction for an example x_i is:

$$y_* = P(y^i = +) = \frac{|\{x' : x' = x^i \wedge y' = +\}|}{|\{x' : x' = x^i\}|}.$$

In conventional bias/variance analysis, x^i records the set of intrinsic attributes about instance i. In relational data, when we use the attributes and class labels of related instances for prediction, our representation for calculating the Bayes-optimal prediction should include more than just the intrinsic attributes of i. However, in real-world relational datasets, when we consider the class labels and attributes of neighboring examples along with the intrinsic attributes of i (i.e., $\{x^i, \mathbf{x}^R, \mathbf{y}^R\}$), there is very little similarity among instances. In fact, in the Cora and WebKB datasets, 100% of examples are unique when we represent an instance i as $\{x_i, \mathbf{x}^R, \mathbf{y}^R\}$ (i.e., $|\{x' : x' = x^i\}| = 1$).

When the relational instances are unique, the Bayes-optimal prediction corresponds to the true value of the class label (i.e., $y_* = \{1 \; \textit{if} \; y^i = +; 0 \; \textit{otherwise}\}$). For the real-data experiments, we use this approximation to compute the learning bias and variance. If the approximation of the Bayes optimal prediction is inaccurate, it can affect estimates of all bias components. The bias measurements of conventional bias/variance methods would be affected similarly. However, in our framework, approximation error can also affect estimates of learning variance because its calculation uses the Bayes-optimal probabilities for neighbors during inference. Since the approximation of Bayes-optimal predictions is only necessary in real datasets, this indicates that researchers should conduct both synthetic and real data experiments to fully investigate model performance.

3.4 Results

3.4.1 Synthetic data

Figure 3 graphs performance on four different types of data. The first set of data has small group size and low linkage, thus we expect it will be difficult for the models to exploit the autocorrelation in the data due to low connectivity. The second set of data has small group size but high linkage, thus we expect the models will be able to exploit neighbor information more effectively. The third set of data has large group size and low linkage. We expect the LGM models to be more accurate on data with large group sizes because they can incorporate information from a wider neighborhood than RDNs and RMNs, which use only local neighbor information. The fourth set of data has large group size and high linkage—we

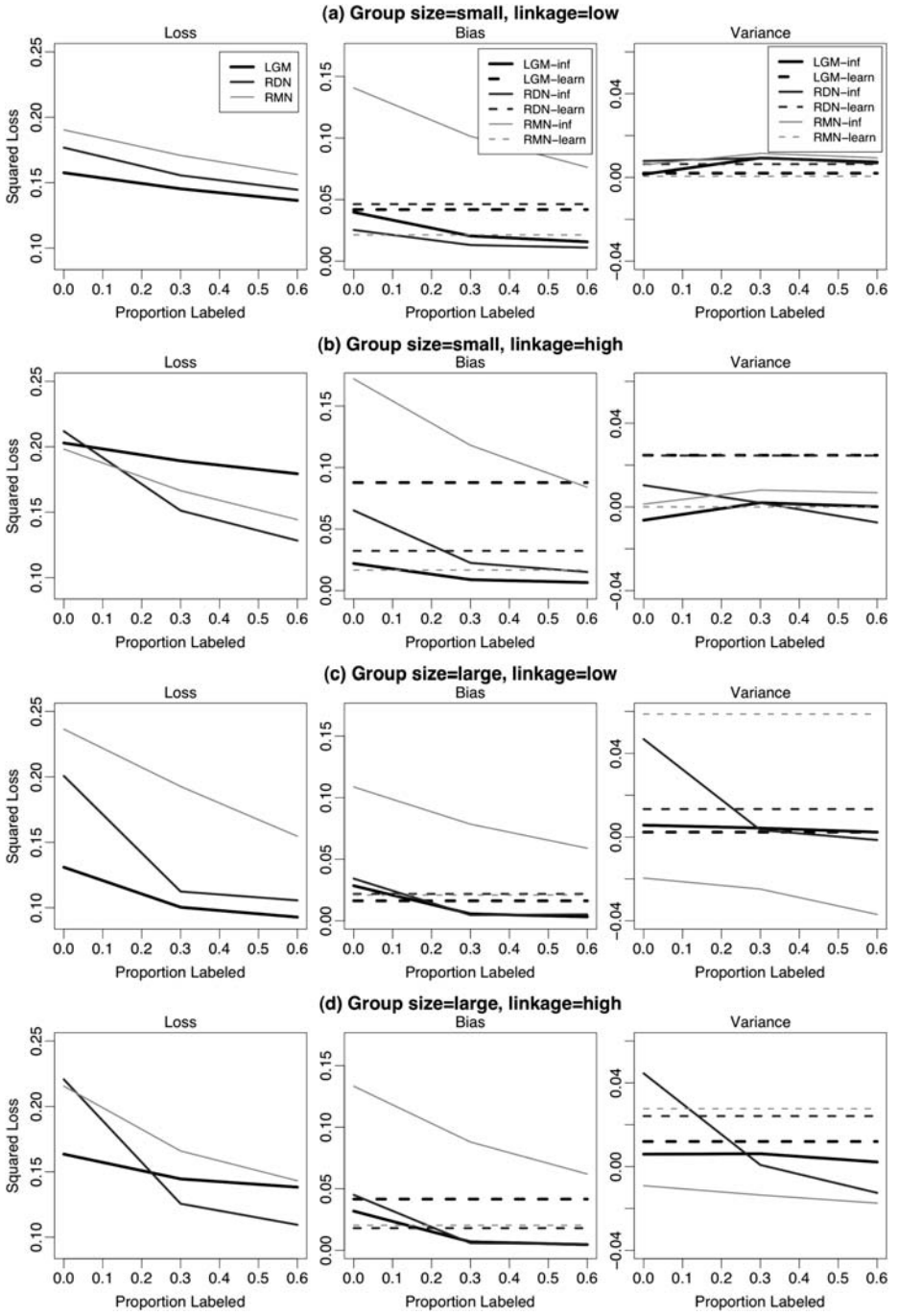


Fig. 3 Bias/variance analysis on synthetic data. Note that the *y*-axes are not aligned

expect the models will be able to exploit autocorrelation dependencies most effectively in these data, due to high connectivity and clustering.

Figure 3 graphs the squared loss decomposition for each model as the level of test-set labeling is varied. When group size is small and linkage is high (row b), LGMs are outperformed by the other two models when the test data are at least partially labeled. The bias/variance decomposition shows that poor LGM performance is due to high learning bias. This is likely due to the LGM algorithm's inability to identify the latent group structure when group size is small and linkage is high. The LGM learning procedure uses a sequential approach where the data are clustered into groups using the link structure alone and the remainder of the model is learned given the identified group structure. When density of linkage between groups is relatively high compared to group size, it will be difficult for the clustering algorithm to correctly identify the fine-grained underlying group structure, and this in turn will bias the learned model. When LGMs are given the true underlying group structure, this bias disappears.

When group size is large and linkage is low (row c), RMNs are outperformed by the other two models regardless of the level of test-set labeling. The bias/variance decomposition shows that poor RMN performance is primarily due to high learning variance. Although the RMN has high inference bias across all four types of data, for this setting it experiences particularly high learning variance. This is somewhat offset by negative inference variance (i.e., inference reduces the overall variance of the predictions) but not completely. This indicates that the RMN learning procedure is overfitting to the training set, which may be due to a larger model space (the combination of larger group size and sparse linkage results in a larger number of possible weight configurations). We experimented with a wide range of priors to limit the impact of overfitting but the effect remained consistent.

The high inference bias exhibited by RMNs is likely due to the use of loopy belief propagation (LBP) for inference. Regardless of our linkage setting, there are likely to be many short cycles in the graphs since the objects are clustered in groups. These short cycles will degrade the quality of LBP inference because they violate its implicit assumption of *tree-like* graphs (i.e., that there are only very long cycles). When LBP is applied to collectively infer the labels in a test set with little seed information, the inference process may converge to extreme autocorrelated labellings (e.g., all positive labels in some regions of the graph, all negative labels in other regions), resulting in high inference bias.

When group size is large and linkage is high (row d), RDNs perform worse than LGMs when there is 0% test-set labeling but perform significantly better when the test data are partially labeled. The bias/variance decomposition shows that poor RDN performance is due to high inference variance, which decreases as labeling increases. The RDN inference algorithm uses Gibbs sampling, seeded with a randomly labeled test set. When there are few labeled objects in the test set, the inference process may be unduly influenced by the initial random labeling of the test set if the RDN model has selected the class label in lieu of other known attributes in the data. When such RDN models are applied to an unlabeled test set, the initial random Gibbs labeling may bias the inference process to converge to widely varying labellings. Thus the initial random labeling can increase the variance of predictions over multiple runs of inference, particularly when there is little information to seed the inference process.

Figure 4 graphs the γ values for each type of synthetic data. Recall that positive values of γ indicate that the mean prediction of the total distribution (y_{Tm}) is farther from the optimal prediction (y_*) than the mean prediction of the learning distribution (y_{Lm}) and that both deviate in the same direction. Negative values of γ indicate that either (1) y_{Tm} and y_{Lm} are on opposite sides of y_*, or (2) y_{Lm} is farther from y_* than y_{Tm} (i.e., inference improves the

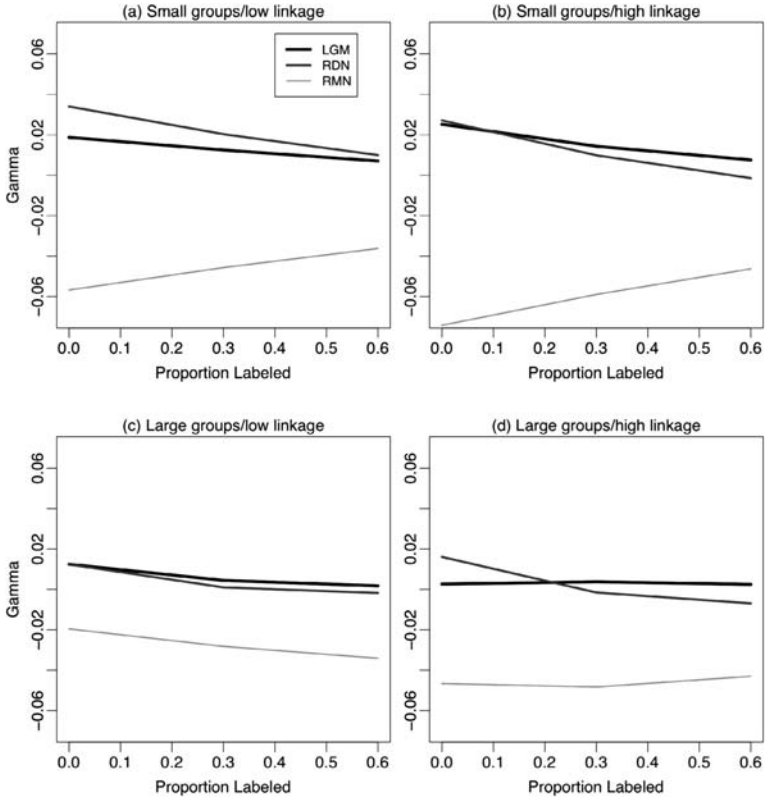


Fig. 4 Synthetic data experiments: γ component

model's predictions). The negative γ values for RMN models are due to the first case. When the model uses the optimal neighbor labels for prediction, the RMN makes one type of error (e.g., undershooting the optimal probability), and when the model uses collective inference for prediction it makes the other type of error (e.g., overshooting the optimal probability). This occurs consistently across the four types of synthetic data that we used for evaluation.

3.4.2 Real-world data

Figure 5 graphs LGM and RDN performance on the WebKB data. On these data the LGM model outperforms the RDN model, most significantly at the 30% level of labeling. This is primarily due to high inference bias. Notice that the LGM inference bias decreases more quickly than the RDN inference bias as labeling increases. This shows how the LGM model is able to more fully exploit the information in a partially labeled test set. Also, in these data the LGM model has consistently lower values of learning bias, inference bias, and learning variance. The LGM only underperforms in terms of inference variance, which is counter to what we observed in the synthetic data experiments when the RDN demonstrated high inference variance. This may indicate that the approximation to the optimal predictions ascribes more variance to the learning component than it should.

Figure 6 graphs LGM and RDN performance on the Cora data. On these data the RDN model outperforms the LGM model at low levels of test-set labeling. This is due to higher

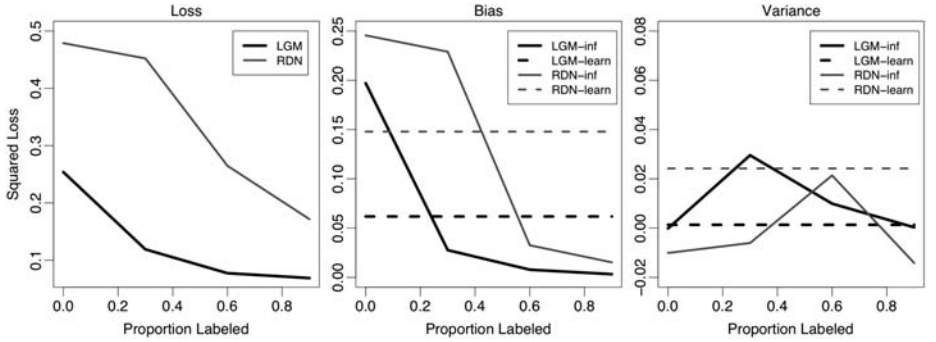


Fig. 5 Bias/variance analysis on WebKB data. Note that the y-axes are not aligned

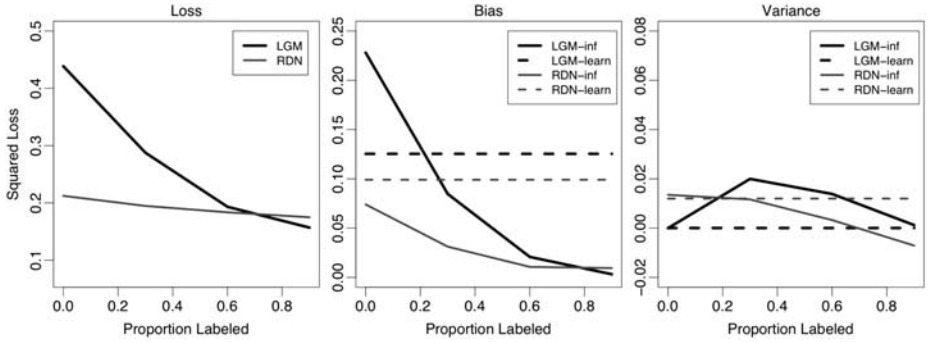


Fig. 6 Bias/variance analysis on Cora data. Note that the y-axes are not aligned

bias—both learning and inference bias. At 0% labeled the inference bias dominates, but at 30% labeled the learning bias dominates. This shows the change in the components of error as the test-set characteristics vary. When there are many labeled objects in the test set, the error components more closely resemble the conventional bias/variance decomposition that attributes error to the learning process alone.

4 Discussion

The synthetic data experiments measure model performance over a range of data characteristics, illustrating the situations in which we can expect each model to perform well. In particular, both the LGM and RDN models perform close to optimal[7] when group size is large and linkage is high (row d). This indicates that as clustering and connectivity increase, the performance of relational models may improve (given moderate levels of autocorrelation).

These experiments have shown several characteristics of relational data that can impact model performance. Graph structure, autocorrelation dependencies, and amount of test-set labeling, all affect relational model performance. LGMs are more robust to sparse labeling

[7]For these datasets, $N_T \cong 0.09$ so the models cannot achieve a squared loss lower than 0.09.

and perform well when graph clustering is high. When the underlying groups are small and linkage is low, LGMs experience high learning bias due to poor cluster identification. RDNs, applied with Gibbs sampling, experience high variance on test data with sparse labeling, but perform well across a wide range of graph structures. RMNs, applied with loopy belief propagation, have higher bias on densely connected graphs, but are more robust to sparse test-set labeling. Our analysis has demonstrated the error introduced by the use of collective inference techniques and how that error varies across models and datasets. This suggests a number of directions to pursue to improve model performance—either by incorporating properties of the inference process into learning or through modification of the inference process based on properties of learning.

These experiments also help us understand model limitations and suggest a number of ways to improve the design of relational learning/inference algorithms. To improve LGM performance, we need to improve the identification of clusters when inter-group linkage drowns out a weak intra-group signal. This may be achieved by the use of alternative clustering techniques in the LGM learning approach, or through the development of a joint learning procedure that clusters for groups while simultaneously estimating attribute dependencies in the model.

To improve RDN performance, we need to improve inference when there are few labeled objects in the test set. This may be achieved through the use of non-random initial labeling to seed the Gibbs sampling procedure. We have started exploring the use of relational probability trees (Neville et al. 2003), learned on the known attributes in the data, to predict class labels for use in the initial Gibbs labeling. Preliminary results indicate that this modification to the inference procedure reduces RDN loss by 10–15% when there is 0% test-set labeling. Alternatively, we could improve the RDN learning algorithm by using meta-knowledge about the test set to bias the feature selection process. For example, if we know that the model will be applied to an unlabeled test set, then we can bias the selective learning procedure to prefer attributes that will be known with certainty during the inference process.

Finally, to improve RMN performance, we need to improve inference when connectivity is high, either when there are large clusters or when overall linkage is dense. This may be achieved through the use of approximate inference techniques other than loopy belief propagation, or through the use of aggregate features in clique templates (that summarize cluster information) rather than using redundant pairwise features. Alternatively, when using pairwise clique templates in a densely connected dataset, it may be helpful to downsample the links in the graph to reduce inference bias.

5 Conclusion

This paper presents a new bias/variance framework that decomposes squared-loss error for model *systems*, which consist of a learning algorithm for model estimation and an inference algorithm used for prediction. To date, work on relational models has focused primarily on the development of models and algorithms rather than the analysis of mechanisms behind model performance. In particular, the impact of collective inference techniques applied to graphs of various structure has not been explored. This work has demonstrated the effects of graph characteristics on relational model performance, illustrating the situations in which we can expect each model to perform well. These experiments also help us understand model limitations and suggest a number of ways to improve the design of relational learning/inference algorithms.

There are a number of ways to improve on our initial work with this framework. First, to facilitate a more extensive analysis of models on real datasets, we are developing (1) relational kernel density estimation techniques to approximate the optimal predictions in a more accurate manner, and (2) relational resampling techniques to generate a more representative set of training samples from a single interconnected training graph. Next, we are examining the interaction effects between learning and inference errors at a local level to inform and guide the design of *joint* learning and inference procedures. Procedures that take the characteristics of inference algorithms into account and bias the learning process are likely to produce more robust and accurate relational models (Wainwright 2005). Finally, we plan to extend the framework to analyze additional aspects of model performance. In particular, the analysis of alternative loss functions (e.g., zero-one) and analysis of errors when estimating the full joint (rather than marginals), will increase our understanding of model performance over a wider range of conditions.

Acknowledgements We thank our anonymous reviewers and Cindy Loiselle for their thoughtful and constructive comments.

This material is based on research sponsored by NSF, DARPA, AFRL, and IARPA under agreement numbers IIS-0326249, HR0011-07-1-0018, HR0011-04-1-0013, and FA8750-07-2-0158. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusion contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of NSF, DARPA, AFRL, IARPA, or the U.S. Government.

References

Bernstein, A., Clearwater, S., & Provost, F. (2003). The relational vector-space model and industry classification. In *Proceedings of the IJCAI-2003 workshop on learning statistical models from relational data* (pp. 8–18).

Craven, M., DiPasquo, D., Freitag, D., McCallum, A., Mitchell, T., Nigam, K., & Slattery, S. (1998). Learning to extract symbolic knowledge from the world wide web. In *Proceedings of the 15th national conference on artificial intelligence* (pp. 509–516).

Domingos, P. (2000). A unified bias-variance decomposition for zero-one and squared loss. In *Proceedings of the 17th national conference on artificial intelligence* (pp. 564–569).

Domingos, P., & Pazzani, M. (1997). On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning*, *29*, 103–130.

Duda, R., Hart, P., & Stork, D. (2001). *Pattern classification*. New York: Wiley.

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

Geman, S., Bienenstock, E., & Doursat, R. (1992). Neural networks and the bias/variance dilemma. *Neural Computation*, *4*, 1–58.

Getoor, L., Friedman, N., Koller, D., & Pfeffer, A. (2001). Learning probabilistic relational models. In *Relational data mining* (pp. 307–335). Berlin: Springer.

Goodman, L. (1961). Snowball Sampling. *Annals of Mathematical Statistics*, *32*, 148–170.

Heckerman, D., Chickering, D., Meek, C., Rounthwaite, R., & Kadie, C. (2000). Dependency networks for inference, collaborative filtering and data visualization. *Journal of Machine Learning Research*, *1*, 49–75.

Hill, S., Provost, F., & Volinsky, C. (2006). Network-based marketing: identifying likely adopters via consumer networks. *Statistical Science*, *22*(2).

Holte, R. (1993). Very simple classification rules perform well on most commonly used datatsets. *Machine Learning*, *11*, 63–91.

James, G. (2003). Variance and bias for general loss functions. *Machine Learning*, *51*, 115–135.

Jensen, D., & Neville, J. (2002). Linkage and autocorrelation cause feature selection bias in relational learning. In *Proceedings of the 19th international conference on machine learning* (pp. 259–266).

Jensen, D., Neville, J., & Gallagher, B. (2004). Why collective inference improves relational classification. In *Proceedings of the 10th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 593–598).

Macskassy, S., & Provost, F. (2007). Classification in networked data: a toolkit and a univariate case study. *Journal of Machine Learning Research*, *8*, 935–983.

McCallum, A., Nigam, K., Rennie, J., & Seymore, K. (1999). A machine learning approach to building domain-specific search engines. In *Proceedings of the 16th international joint conference on artificial intelligence* (pp. 662–667).

Murphy, K., Weiss, Y., & Jordan, M. (1999). Loopy belief propagation for approximate inference: an empirical study. In *Proceedings of the 15th conference on uncertainty in artificial intelligence* (pp. 467–479).

Neville, J., & Jensen, D. (2004). Dependency networks for relational data. In *Proceedings of the 4th IEEE international conference on data mining* (pp. 170–177).

Neville, J., & Jensen, D. (2005). Leveraging relational autocorrelation with latent group models. In *Proceedings of the 5th IEEE international conference on data mining* (pp. 322–329).

Neville, J., Jensen, D., Friedland, L., & Hay, M. (2003). Learning relational probability trees. In *Proceedings of the 9th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 625–630).

Taskar, B., Abbeel, P., & Koller, D. (2002). Discriminative probabilistic models for relational data. In *Proceedings of the 18th conference on uncertainty in artificial intelligence* (pp. 485–492).

Wainwright, M. (2005). Estimating the "wrong" Markov random field: benefits in the computation-limited setting. In *Advances in neural information processing systems*.