Bias/Variance Analysis for Relational Domains

Jennifer Neville[1] and David Jensen[2]

[1] Departments of Computer Science and Statistics,
Purdue University, West Lafayette, IN 47907-2107, USA
[2] Department of Computer Science
University of Massachusetts Amherst, Amherst, MA 01003-4610, USA

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 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. In this paper, we propose a new bias/variance framework that decomposes loss into errors due to both the *learning* and *inference* process. We evaluate performance of three relational models 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.

Key words: Statistical relational learning, collective inference, evaluation

1 Introduction

Bias/variance analysis (e.g., [3]) has been used for a number of years to investigate the mechanisms behind model performance. This analysis is based on the fundamental understanding that prediction error has two components (bias and variance) and that there is often a tradeoff between the two 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 (e.g., [7]).

Conventional bias/variance analysis decomposes loss into errors due to aspects of learning procedures. Loss is decomposed into three factors: bias, variance and noise. In the traditional decomposition, bias and variance measure estimation errors in 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 [2].

The assumption underlying the conventional decomposition is that there is no variation in model predictions due to (1) the inference process, and (2) the

available information in the test set. Classification of relational data often violates these assumptions when *collective inference* techniques are used. If there are dependencies among the class labels of related instances, the inferences about one object can be used to improve the inferences about other related objects. Collective inference techniques exploit these dependencies by simultaneously inferring values over the entire dataset and results in more accurate predictions than conditional inference for each instance independently [8].

Collective inference often requires the use of approximate inference techniques, which may introduce variation in model predictions for a single instance. For example, final predictions for an instance may depend on the initial (random) start state used during inference, thus multiple runs of inference may result in different predictions. 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. Current methods for evaluating relational learning techniques typically assume that labeling different nodes in the test set have equivalent impact. However, the heterogeneity of the relational graph may allow some instances to have more of an impact on neighbor predictions than others—thus, *which* instances are labeled in the test set may cause additional variation in the predictions. Finally, relational models are generally learned on a fully labeled training set (i.e., the class labels of all neighbors are known), but then applied to an unlabeled, or partially labeled, test set. This mismatch between training and test set information may impact the final model predictions.

To date, the impact of *inference* error on model performance has not been investigated. In this paper, we propose a new bias/variance framework that decomposes marginal squared-loss error into components of both the *learning* and *inference* process. We evaluate performance of three relational models on synthetic data and use 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 [12] have higher inference bias in densely connected networks; relational dependency networks [9] have higher inference variance when there is little information to seed the inference process; latent group models [10] 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 models' performance.

2 Framework

In conventional bias/variance analysis, loss is decomposed into three factors: bias, variance and noise [3, 1]. 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 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 optimal prediction y_*: $B(x) = L(y_*, y_m)$. 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 estimates are typically calculated 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 *learning*, not aspects of *inference*. The technique assumes that exact inference is possible and that the training and test sets have the same available information. However, in relational datasets there can be additional variation due to the use of approximate 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 the inference process.

For relational data, we first define the expected *total* loss for an instance x as an expectation over training sets D_{tr} and test sets D_{te}. Following the standard decomposition for loss as described in [4], we can decompose *total* loss into *total* bias, variance, and noise:

$$
\begin{aligned}
&E_{Dtr,Dte,t}[L(t,y)] \\
&= E_{Dtr,Dte,t}[(t-y)^2] \\
&= E_t[(t-E[t])^2] + E_{Dtr,Dte}[(y-E[t])^2] \\
&= N_T(x) + E_{Dtr,Dte}[(y - E_{Dtr,Dte}[y] + E_{Dtr,Dte}[y] - E[t])^2] \\
&= N_T(x) + E_{Dtr,Dte}[(y - E_{Dtr,Dte}[y])^2 + (E_{Dtr,Dte}[y] - E[t])^2 + \\
&\quad 2(y - E_{Dtr,Dte}[y]) \cdot (E_{Dtr,Dte}[y] - E[t])] \\
&= N_T(x) + E_{Dtr,Dte}[(y - E_{Dtr,Dte}[y])^2] + (E_{Dtr,Dte}[y] - E[t])^2 \\
&= N_T(x) + V_T(x) + B_T(x)
\end{aligned}
$$

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 algorithms.

Then we define the *learning* loss as an expectation over training sets D_{tr} alone, using a fully labeled test set for inference. For example, when predicting the class label for instance x_i, the model is allowed to use the class labels (and attributes) of all other instances in the dataset ($X - \{x_i\}$). This enables the application of exact inference techniques 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. For this reason, we will refer to the components as *learning* bias, variance, and noise:

$$
\begin{aligned}
&E_{Dtr,t}[L(t,y)] \\
&= E_{Dtr,t}[(t-y)^2] \\
&= E_t[(t-E[t])^2] + E_{Dtr}[(y-E[t])^2] \\
&= N_L(x) + E_{Dtr}[(y-E_{Dtr}[y]+E_{Dtr}[y]-E[t])^2] \\
&= N_L(x) + E_{Dtr}[(y-E_{Dtr}[y])^2 + (E_{Dtr}[y]-E[t])^2 + \\
&\quad 2(y-E_{Dtr}[y]) \cdot (E_{Dtr}[y]-E[t])] \\
&= N_L(x) + E_{Dtr}[(y-E_{Dtr}[y])^2] + (E_{Dtr}[y]-E[t])^2 \\
&= N_L(x) + V_L(x) + B_L(x)
\end{aligned}
$$

Once we have measured the *total* and *learning* bias/variance, we can define *inference* bias/variance as the difference between the total error and the error due to the learning process alone:

$$
\begin{aligned}
B_I(x) &= B_T(x) - B_L(x) \\
V_I(x) &= V_T(x) - V_L(x)
\end{aligned}
$$

For example, consider the distributions of model predictions in figure 1. We measure the variation of model predictions for an instance x in two ways. First, when we generate synthetic data we 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 class labels of related instances to be used during 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 a single test set. These predictions form the *total* distribution, with a mean *total* prediction of y_{Tm}. The model's *learning* bias is calculated as the difference between y^* and y_{Lm}; the *inference* bias is calculated as the difference between y_{Lm} and y_{Tm}. The model's *learning* variance is calculated from the spread of the *learning* distribution; the *inference* variance is calculated as the difference between the *total* variance and the *learning* variance.

3 Experiments

To explore the effects of relational graph and attribute structure on model performance, we generated synthetic datasets with varying levels of autocorrelation, linkage, and group structure. Group structure is used to control the inherent clustering of the data. Our experiments evaluate model performance in a classification context, where only a single attribute is unobserved in the test set. We generated data in the manner described below, and learned models to predict X_1 using the intrinsic attributes of the object (X_2, X_3, X_4) as well as the class label and the attributes of directly related objects (X_1, X_2, X_3, X_4). We evaluated three relational models, measuring squared loss and decomposing it into bias and variance components for each model.

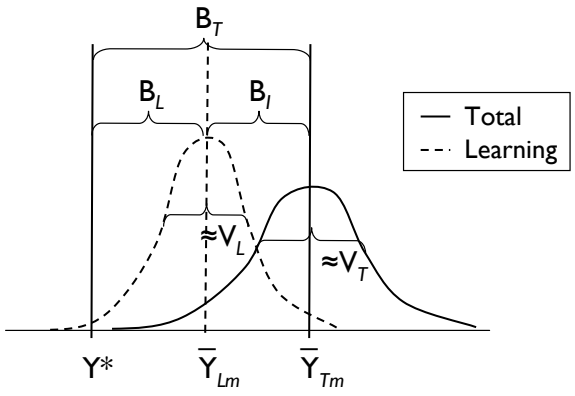

Fig. 1. Distributions of model predictions.

3.1 Synthetic data

Our synthetic datasets are homogeneous data graphs with autocorrelation due to an underlying (hidden) group structure. Each object has a group G and four boolean attributes: X_1, X_2, X_3 and X_4. Each group has an associated type T. We used the generative process described in Table 1 to generate a dataset with N_O objects and G_S average group size, using the settings specified below. The procedure uses a simple model where X_1 has an autocorrelation level of 0.5[3], X_2 depends on X_1, and the other two attributes have no dependencies.

$$
\begin{aligned}
N_O &= 250 \\
p(T) &= \{p(T{=}1) = 0.50; p(T{=}0) = 0.50\} \\
p(X_1|T_G) &= p(X_1{=}1|T_G{=}1){=}0.90; p(X_1{=}0|T_G{=}0){=}0.90. \\
p(X_2|X_1) &= p(X_2{=}1|X_1{=}1){=}0.75; p(X_2{=}0|X_1{=}0){=}0.75. \\
p(X_3{=}1) &= p(X_4{=}1) = 0.50
\end{aligned}
$$

We generated data with two different groups sizes and levels of linkage:

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

[3] 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.

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.

Table 1. Synthetic data generation

For each group g, $1 \leq g \leq (N_G = N_O/G_S)$:
Choose a value for group type t_g from $p(T)$.
For each object i, $1 \leq i \leq N_O$:
Choose a group g_i uniformly in $[1, N_G]$.
Choose a class value X_{1i} from $p(X_1 \mid T_{G_i})$.
Choose a value for X_{2i} from $p(X_2 \mid X_1)$.
Choose values for X_{3i} from $p(X_3)$ and X_{4i} from $p(X_4)$.
For each object j, $1 \leq j \leq N_O$:
For each object k, $j < k \leq N_O$:
Choose whether the two objects are linked from $p(E \mid G_j = G_k)$.

3.2 Models

We compare the performance of three different relational models: relational Markov networks (RMNs) [12], relational dependency networks (RDNs) [9], and latent group models (LGMs) [10].

RMNs 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 a 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 posterior 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.

RDNs extend dependency networks [6] 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) [11] as the component CPD to model X_1. Note that the RPT is a selective model (i.e., the learning algorithm select which features are relevant to the task), so it may not use all the available attributes. For inference,

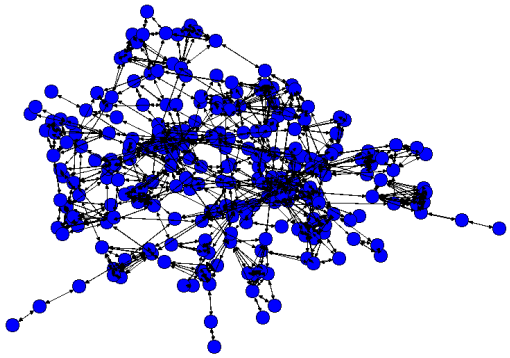

Fig. 2. Sample synthetic dataset.

we used Gibbs sampling with fixed-length chains of 2000 samples and a burn-in length of 100.

LGMs 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 [5], link existence uncertainty, and hierarchical latent variables. The model posits groups of objects in the data of various type. 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.

3.3 Results

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 model performance 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.

To measure the expected loss over training and test sets, we used the following procedure:

1. For each outer trial $i = [1, 5]$:
   (a) Generate test set.
   (b) For each learning trial $j = [1, 5]$:

i. Generate training set, record optimal predictions.
ii. Learn model of X_1 on the training set.
iii. Infer marginal probabilities for test set with fully labeled test data (i.e., $\mathbf{X} - \{X_i\}$), record *learning* predictions.
iv. For each inference trial $k = [1, 5]$ and proportion labeled $p = [0.0, 0.3, 0.6]$:
A. Randomly label $p\%$ of test set.
B. Infer marginal probabilities for unlabeled test instances, record *total* predictions.
C. Measure squared loss.

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

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

Figure 3 graphs performance on four different types of data. The first set of data have 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 have 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 have 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 have large group size and high linkage—we 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 RDNs 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), LGMs significantly outperform RDNs when there is 0% test set labeling. The bias/variance decomposition shows that poor RDN performance is due to high inference variance. (Note the difference between RDN total variance and learning variance.) The RDN inference algorithm uses Gibbs sampling, seeded with a randomly labeled test set. When there are few labeled instances 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 labelings. 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.

When group size is large and linkage is high (row d), LGMs outperform RMNs regardless of the level of test set labeling. The bias/variance decomposition shows that poor RMN performance is due to high inference bias. (Note the difference between RMN total bias and RMN learning bias.) This indicates that the RMN inference procedure is likely to bias the marginal probability estimates when run in a densely connected network with little seed information. This may be due to the algorithm learning skewed clique weights on a fully labeled training set. When these weights are applied to collectively infer the labels throughout the test set, the inference process may converge to extreme labelings (e.g., all positive labels in some regions of the graph, all negative labels in other regions) when the graph is very "loopy" (i.e., densely connected). We experimented with a wide range of priors to limit to the impact of weight overfitting but the effect remained consistent.

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[4] 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 the relational data characteristics 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 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.

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

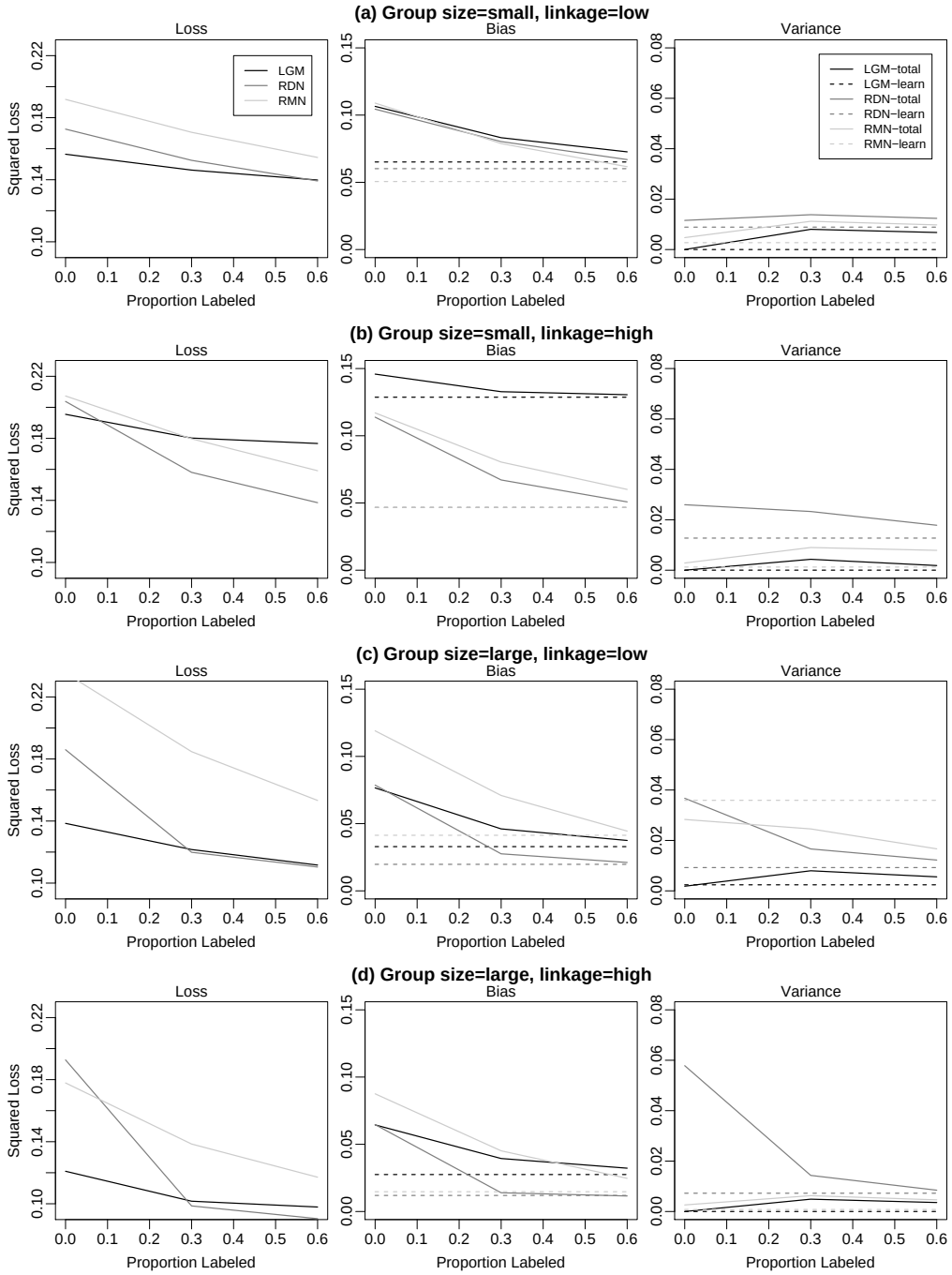

Fig. 3. Bias/variance analysis on synthetic data.

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 instances 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 relational probability trees [11], 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 into aspects of both the *learning* and *inference* processes. To date, work on relational models 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 two ways to improve on our initial work with this framework. First, we intend to broaden our analysis to real data sets and evaluate algorithm modifications in these domains. This will lead towards a full characterization of the situations in which we can expect relational models to achieve superior performance. Next, 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. Also, examining interaction effects between learning and inference errors may help to inform the design of joint learning and inference procedures, which could significantly extend the performance gains of relational models.

Acknowledgments

This research is supported by DARPA and NSF under contract numbers HR0011-04-1-0013 and IIS-0326249. 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, NSF, or the U.S. Government.

References

1. P. Domingos. A unified bias-variance decomposition for zero-one and squared loss. In *Proceedings of the 17th National Conference on Artificial Intelligence*, pages 564–569, 2000.
2. P. Domingos and M. Pazzani. On the optimality of the simple bayesian classifier under zero-one loss. *Machine Learning*, 29:103–130, 1997.
3. J. Friedman. On bias, variance, 0/1-loss, and the curse-of-dimensionality. *Data Mining and Knowledge Discovery*, 1(1):55–77, 1997.
4. S. Geman, E. Bienenstock, and R. Doursat. Neural networks and the bias/variance dilemma. *Neural Computation*, 4:1–58, 1992.
5. L. Getoor, N. Friedman, D. Koller, and A. Pfeffer. Learning probabilistic relational models. In *Relational Data Mining*, pages 307–335. Springer-Verlag, 2001.
6. D. Heckerman, D. Chickering, C. Meek, R. Rounthwaite, and C. Kadie. Dependency networks for inference, collaborative filtering and data visualization. *Journal of Machine Learning Research*, 1:49–75, 2000.
7. R. Holte. Very simple classification rules perform well on most commonly used datatsets. *Machine Learning*, 11:63–91, 1993.
8. D. Jensen, J. Neville, and B. Gallagher. Why collective inference improves relational classification. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 593–598, 2004.
9. J. Neville and D. Jensen. Dependency networks for relational data. In *Proceedings of the 4th IEEE International Conference on Data Mining*, pages 170–177, 2004.
10. J. Neville and D. Jensen. Leveraging relational autocorrelation with latent group models. In *Proceedings of the 5th IEEE International Conference on Data Mining*, pages 322–329, 2005.
11. J. Neville, D. Jensen, L. Friedland, and M. Hay. Learning relational probability trees. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 625–630, 2003.
12. B. Taskar, P. Abbeel, and D. Koller. Discriminative probabilistic models for relational data. In *Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence*, pages 485–492, 2002.