International Journal of Advances in Engineering Research http://www.ijaer.com

(IJAER) 2013, Vol. No. 5, Issue No. II, February **ISSN: 2231-5152**

ANALYSING COLLECTIVE INTERFERENCE ON THE BASIS OF INFERENCE ACCURACY

ADIL ZAMIL, #DR. RAJESH PATHAK

#HOD Deptt. Of Computer Science, & Engg.
GNIT Institute of Technology,
Gautam Budh Nagar, U.P.

ABSTRACT

For all models except R2, bias quickly becomes nearly level. The bias of R2 continues to decline as data slowly accumulate in joint distributions and overcome the prior (an initial Laplace correction to prevent zero probabilities). Variance is level for Intrinsic and CI, but continues to decline for R2, and less so for R1 and RCI, between Train Size values of 1000 and 5000. For all points with Train Size > 100, the 95% confidence intervals are less than 0.002.

Thus, the large initial gap in loss between CI and R2 appears directly attributable to the size of R2's parameter space, and the difficulty of making good estimates with sparse data. This difference between CI and R2 is pronounced even though objects in the data contain only three attributes. If the number of attributes on each object is increased, as shown in figure 5, the loss of R2 soars compared to other models, including CI, whose loss remains nearly constant. Even R1 and RCI show marked increases in loss, though to a much smaller extent than R2. The results for data produced by collective generation are qualitatively similar.

Key words: *decline, probabilities, attributable, pronounced., qualitatively.*

INTRODUCTION

In choosing a representation, we must keep in mind the five issues: complexity, uncertainty, modularity, comprehensibility, and inference. The first two suggest a method that combines first-order logic with probability. There are many such methods. One of them, SLPs, assumes that for a given consequent, only one rule can fire at a time. SLPs are thus not a natural fit when multiple rules can function simultaneously as sources of evidence for their common consequent. Another method, probabilistic relational models, lacks the modularity required for construction by many loosely-coordinated individuals. The various relational methods available today are mostly not as modular or comprehensible as, say, a collection of first-order rules.

Further, most of them did not yet exist at the time we were creating this architecture. KBMC, on the other hand, fulfills all the desiderata. Horn clauses have the key feature of high modularity: a new rule can be input without knowing what other rules are already in the knowledge base. Horn rules are also very comprehensible: rules can be read as "if-then" statements, making them natural for reading and writing. The only apparent drawback to using

KBMC is its use of Horn clauses instead of full first-order logic. In their defense, however, Horn clauses are used in many expert system shells, and form the basis of the Prolog programming language. They Collective Knowledge Base Users Rules Facts Feedback

REVIEW OF LITERATURE

In current knowledge-sharing sites and knowledge management systems, questions are answered from an indexed repository of past answers, or routed to the appropriate experts. Thus the only questions that can be answered automatically are those that have been asked and answered in the past. In contrast, the architecture we propose here allows chaining between rules and facts provided by different experts, and thus automatically answering potentially a very large number of questions that were not answered before. This can greatly increase the utility of the system, decrease the cost of answering questions, and increase the rewards of contributing knowledge.

The architecture answers the problems posed in the introduction:

Quality:- By employing feedback and machine learning, we are able to determine which rules are of high quality, and which are not. Further, since we are tracking the utility of knowledge provided by users, they are more inclined to provide good rules.

Consistency:- By using a probabilistic framework, we are able to handle inconsistent knowledge.

Relevance :- Since the knowledge base is being built by users, for users, we expect the rules to be on topics that the users find relevant and interesting. The credit assignment process rewards those contributors whose rules are used (and produce correct answers), which provides incentive to create rules that are relevant to users' needs.

Scalability:- For both training and query-answering, the most expensive portion of the computation is the probabilistic inference on the Bayesian network. However, this computation depends only on the size of the network, not of the entire knowledge base. The Bayesian network is constructed out of only the relevant knowledge, which we expect (and confirm empirically in the experimental section) will typically lead to relatively small networks even for very large knowledge bases.

Motivation of contributors:- By tracking the utility of rules and assigning credit to those which are used to answer queries, we provide the means for motivating contributors (e.g. listing the top ten, paying in some real or virtual currency, etc.)

MATERIAL AND METHOD

Squared loss as a function of training set size for all models. Data for figure 3a were produced using the collective generator, so it is not unexpected that CI rapidly converges to a relatively low loss. R2 continues to reduce its loss as Train Size increases. At Train Size=5000, none of the models achieve minimum error, corresponding to the CI model provided with perfect class information, but R2 continues to reduce loss at a steady rate.

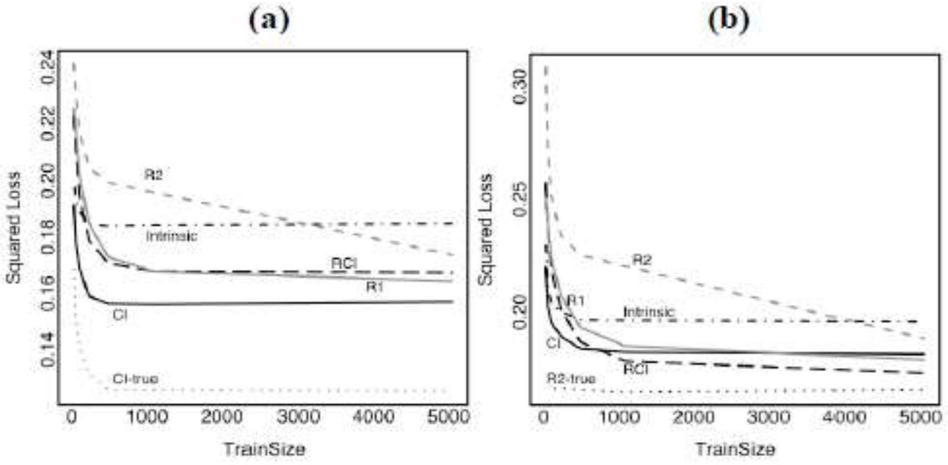


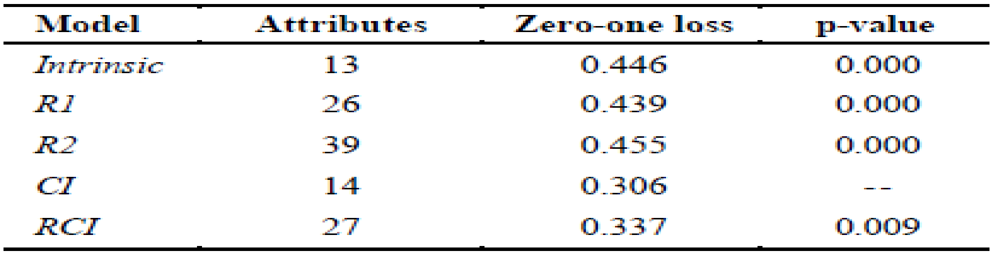
Figure 1: squared loss for (a) collective data generation, and (b) relational data generation

For all points with TrainSize > 100, the 95% confidence intervals are less than 0.004 Figure 3b shows the same type of results for the relational data generator. Even though they do not match the data generator, CI and RCI outperform other models when TrainSize is small.

R2 continues to have corresponding high loss, though it will eventually drop below CI and RCI as it declines to the optimal value of loss at high TrainSize. R1 performs similarly, dropping below CI at around TrainSize=3000. For TrainSize > 100, the 95% confidence intervals are less than 0.004.

We obtained similar results with experiments on the yeast protein data. CI resulted in the lowest zero one loss and its loss was significantly different than the zero one loss of all other models.

Table 1: Yeast Protein Data Zero-One Loss Results



Model	Attributes	Zero-one loss	p-value
Intrinsic	13	0.446	0.000
R1	26	0.439	0.000
R2	39	0.455	0.000
CI	14	0.306	--
RCI	27	0.337	0.009

What is responsible for the low error of CI models? We measured bias and variance for the probability estimates of each model to compare their decomposed loss as a function of Train Size. Figures 4a and b show the results for the relational generator. For the collective generator, the variance results were qualitatively similar and the bias varied only slightly across the range of Train Size.

For all models except R2, bias quickly becomes nearly level. The bias of R2 continues to decline as data slowly accumulate in joint distributions and overcome the prior (an initial Laplace correction to prevent zero probabilities). Variance is level for Intrinsic and CI, but continues to decline for R2, and less so for R1 and RCI, between Train Size values of 1000 and 5000. For all points with Train Size > 100, the 95% confidence intervals are less than 0.002.

Thus, the large initial gap in loss between CI and R2 appears directly attributable to the size of R2's parameter space, and the difficulty of making good estimates with sparse data. This difference between CI and R2 is pronounced even though objects in the data contain only three attributes. If the number of attributes on each object is increased, as shown in figure 5, the loss of R2 soars compared to other models, including CI, whose loss remains nearly constant. Even R1 and RCI show marked increases in loss, though to a much smaller extent than R2. The results for data produced by collective generation are qualitatively similar.

This growth in the number of attributes is modest compared to some of the most common applications of relational learning algorithms, such as classifying web pages, in which objects have hundreds or thousands of attributes (e.g., words on a web page). For these applications, the ability of CI to provide a built-in factoring of the feature space may be almost essential.

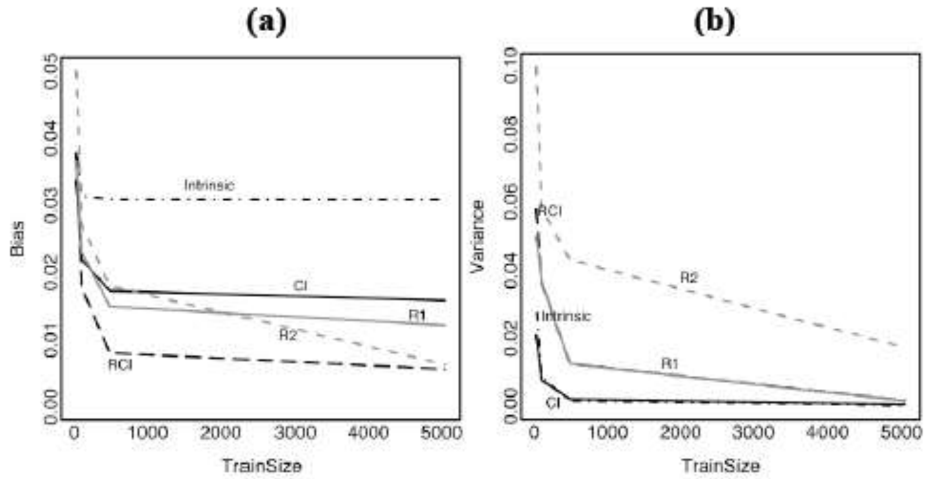


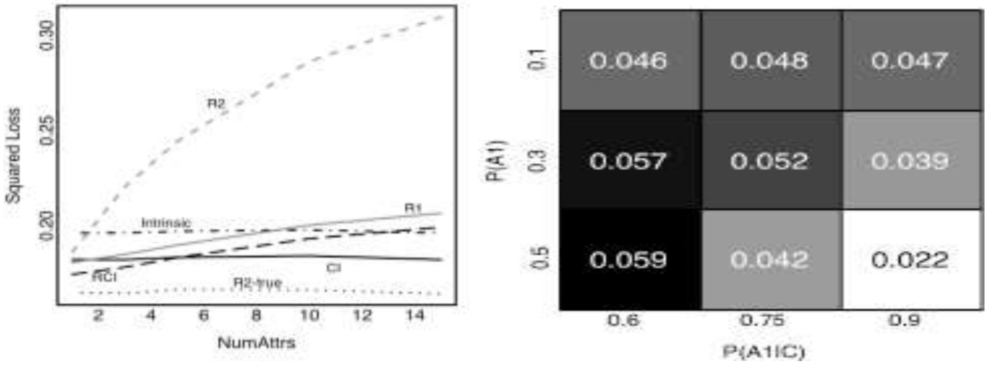
Figure 2 loss decomposition into (a) bias, and (b) variance, fr relational data generation.

The negative effect of large parameter spaces on R2 and RCI is a plausible explanation for the results of our experiments with the yeast protein data. Both R2 and RCI perform worse than CI.

Strength of Probabilistic Dependence

Our data generation procedures use two parameters P(A1) and P(C|A1) to determine the strength of probabilistic dependence between the attribute A1 and the class label C. The relative performance of models differs based on the strength of this dependence. Figure 6 depicts how the quantity loss(R2)–loss(CI) varies as a function of P(A1) and P(C|A1).

The largest difference between the two models occurs when P(A1) is uniformly distributed and the dependence of A and C is weakest. That is, CI performs best, in relative terms, when few correlations exist other than autocorrelation of class labels. The relative advantage of CI disappears as more information is available to R2. However, if no attributes are useful then only CI would be able to attain non-random performance.



Relational autocorrelation refers to the correlation among the values of the same variable on several related objects. The widespread occurrence of autocorrelation is one of the strongest motivations for relational inference of any kind. Its effects have been noted and explored by several researchers in collective inference, including Macskassy & Provost, Taskar et al. , and Yang et al.

Figure shows the effect of increasing levels of autocorrelation on the relative performance of different models.

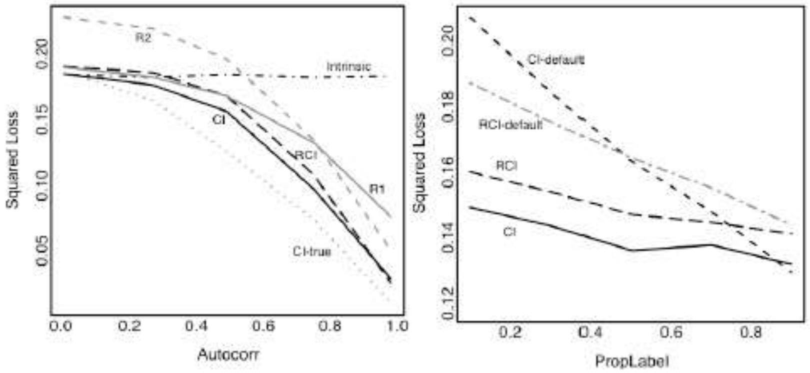


Figure 3 loss as a function of autocorrelation figure 3 loss as a function of percentage of data labeled

All models we consider, except Intrinsic, are greatly aided by autocorrelation, though their relative ordering changes slightly. R1 is aided least by autocorrelation while R2 is aided most.

These results reveal another advantage of CI. Even when autocorrelation is entirely absent, CI's performance is equal to that of Intrinsic. CI can exploit autocorrelation when present, but is not significantly impaired by its absence.

Proportion of Known Values

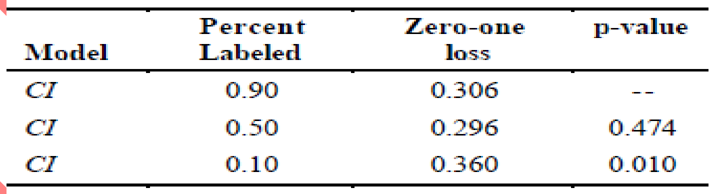
The core of collective inference is that inferences about one object can inform inferences about another. This capability is particularly useful when some values are known with certainty. For example, predictions about the topic of a previously unvisited webpage may be aided by considering the known topics of previously visited pages.

Figure 3 shows how varying the proportion of labeled data affects CI and RCI, and how their performance compares to an alternative inference scheme for these models (labeled default). Rather than conducting full collective inference, default models terminate inference after the first round of Gibbs sampling. These results indicate the advantage of collective inference over non-collective, holding all other factors constant. As shown in figure 8, the relative advantage of collective inference is reduced as more of the data are labeled. That is, collective inference procedures become less and less necessary as the percentage of true labels increases.

While only a few studies [1,9,16] have actively varied the percentage of known labels, the results above closely parallel those of Macskassy and Provost [9], who show that the relative advantage of an iterative inference procedure over a noniterative procedure reduces as the percentage of labeled data increases. They show that, in general, their collective inference procedure performs better when class skew is present or when few labels are known with certainty.

We also evaluated this effect using the yeast protein data, obtaining the results shown in table1. For each of the ten-fold cross-validation partitions, we learned a CI model on the 90% training partition and applied the model to the entire dataset. During collective inference, we varied the proportion of the data that was labeled—the test partition was always unlabeled but the training partition was labeled at the following levels {1.0,0.55,0.11} to produce overall levels of {0.9,0.5,0.1}. Accuracy was measured on the unlabeled instances and averaged over the ten folds. Loss increases significantly when only 10% of the data are labeled, but there is no significant difference in performance between 50% and 90% labeled.

Table 2: Yeast protein data results with. partial labeling



Model	Percent Labeled	Zero-one loss	p-value
CI	0.90	0.306	--
CI	0.50	0.296	0.474
CI	0.10	0.360	0.010

CONCLUSIONS

Our experiments with real and synthetic data indicate that the reduced error attributed to collective inference results primarily from a clever factoring of the space of statistical dependencies in relational data. Models that represent this factoring, when combined with algorithms for collective inference, can greatly reduce bias in data with strong autocorrelation with the minimum possible increase in variance. When autocorrelation is absent, the models have practically equivalent error to their non-relational counterparts.

REFERENCES

- *Abdul-Rahman and S. Hailes, "A distributed trust model," in Proceedings of New Security Paradigms Workshop, 1997, pp. 48–60.*

- *R. Agrawal, S. Dar, and H. V. Jagadish, "Direct transitive closure algorithms: Design and performance evaluation." ACM Transactions on Database Systems, vol. 15, no. 3, pp. 427– 458, 1990.*

- *R. Agrawal and H. V. Jagadish, "Multiprocessor transitive closure algorithms." in Proceedings of the International Symposium on Databases in Parallel and Distributed Systems, Austin, TX, 1988, pp. 56–66.*

- *Agresti, Categorical Data Analysis. New York, NY: Wiley, 1990.*

- *V. Aho, J. E. Hopcroft, and J. D. Ullman, The Design and Analysis of Computer Algorithms. Reading, MA: Addison-Wesley, 1974.*

- *Anderson, P. Domingos, and D. Weld, "Relational Markov models and their application to adaptive Web navigation," in Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Edmonton, Canada: ACM Press, 2002, pp. 143–152.*

- *F. Bacchus, Representing and Reasoning with Probabilistic Knowledge. Cambridge, MA: MIT Press, 1990.*

- *F. Bancilhon, "Naive evaluation of recursively defined relations." in On Knowledge Base Management Systems (Islamorada), 1985, pp. 165–178.*

- *R. Bellman and M. Giertz, "On the analytic formalism of the theory of fuzzy sets." Information Sciences, vol. 5, pp. 149–156, 1973.*

- *T. Berners-Lee, J. Hendler, and O. Lassila, "The Semantic Web," Scientific American, vol. 284, no. 5, pp. 34–43, 2001.*

- *J. Besag, "Statistical analysis of non-lattice data," The Statistician, vol. 24, pp. 179–195, 1975.*

- *M. Blaze, J. Feigenbaum, and J. Lacy, "Decentralized trust management," in Proceedings of the 1996 IEEE Symposium on Security and Privacy, Oakland, CA, 1996, pp. 164–173.*

- *L. Breiman, "Bagging predictors," Machine Learning, vol. 24, pp. 93–140, 1996.*
- *S. Brin and L. Page, "The anatomy of a large-scale hypertextual Web search engine," in Proceedings of the Seventh International World Wide Web Conference. Brisbane, Australia: Elsevier, 1998.*

- *Carre, Graphs and Networks. Oxford: Claredon Press, 1978.*

- *S. Chakrabarti, B. Dom, D. Gibson, J. Kleinberg, P. Raghavan, and S. Rajagopalan, "Automatic resource compilation by analyzing hyperlink structure and associated text," in Proceedings of the Seventh International World Wide Web Conference. Brisbane, Australia: Elsevier, 1998, pp. 65–74.*
-
- *M. Chickering and D. Heckerman, "Efficient approximations for the marginal likelihood of Bayesian networks with hidden variables," Machine Learning, vol. 29, pp. 181–29, 1997.*

- *K. Chow and C. N. Liu, "Approximating discrete probability distributions with dependence trees," IEEE Transactions on Information Theory, vol. 14, pp. 462–467, 1968.*

- *S. A. Cook, "The complexity of theorem-proving procedures," in Proceedings of the Third Annual ACM Symposium on Theory of COmputing, 1971, pp. 151–158. [Online]. Available: http://theory.lcs.mit.edu/ dmjones/STOC/stoc71.html*

- *G. F. Cooper, "The computational complexity of probabilistic inference using Bayesian belief networks." Artificial Intelligence, vol. 42, no. 2-3, pp. 393–405, 1990.*

- *V. S. Costa, D. Page, M. Qazi, , and J. Cussens, "CLP(BN): Constraint logic programming for probabilistic knowledge," in Proceedings of the Nineteenth Conference on Uncertainty in Artificial Intelligence. Acapulco, Mexico: Morgan Kaufmann, 2003, pp. 517–524.*