A New Method for Estimating Causal Model Learning Accuracy

Erich Kummerfeld, PhD[1], Gregory F. Cooper, MD, PhD[2]
[1]University of Minnesota, Minneapolis, MN; [2]University of Pittsburgh, Pittsburgh, PA

Abstract

After learning a causal model by running a causal discovery algorithm on a real world data set, investigators would like to estimate the accuracy of the learned causal model. It is difficult to obtain an accurate estimate, however, since knowledge of the causal truth, which serves as the gold standard, is often incomplete and sometimes incorrect. One method that is sometimes used to estimate accuracy is resimulation. Resimulation is a process in which data are generated (simulated) from the learned model. An accuracy estimate is then obtained by running the discovery algorithm on this resimulated data set, and comparing its output model against the model that was used to generate the resimulated data. This paper introduces a new method called hybrid resimulation (Hsim) that estimates causal learning accuracy by using a learning dataset that contains both real and resimulated data. In a simulation study we show the difficulty of graph accuracy estimation, and we compare the performance of Hsim to that of standard full resimulation. The results support that Hsim provides a better estimate of algorithm accuracy when the underlying causal mechanisms are nonlinear. We also compare these methods in a case study using the Breast Cancer Wisconsin (Diagnostic) Data Set.

Introduction

After learning a causal model by applying a causal discovery algorithm to a real world data set, investigators would like to estimate the accuracy of the learned causal model. Often the true causal relationships are not known completely and accurately. A simple method of evaluation is to use benchmark simulation studies, often published alongside the causal discovery methods themselves[1,2]. These benchmark simulation studies report various performance measures when the algorithms are applied to simulated data that have particular sample sizes, numbers and types of variables, generating graphical structures, functional relationships, and parameter values.

There are limitations to this standard simulation-based approach, however, due to the richness and complexity of real world data. While in many cases it should be feasible to find simulation studies with approximately the same number of samples and variables, the other model features—graphical structure, functional relationships, parameter values—might vary significantly between the real world data and the simulations. This would not be a problem if causal discovery methods were not sensitive to differences in these model features, but they are. To make matters worse, an appropriate benchmark simulation study might not only be unavailable, but unknown; the features of real world data sets may not be known or known with sufficient precision, including for example feedback, temporal dynamics, non-linearity, mixtures of distributions, and many others.

This paper introduces a different method for estimating accuracy, namely, a hybrid method that combines real data and resimulated data[3]. Resimulation is a process in which data are generated (simulated) from a learned model. An accuracy estimate is then obtained by running the discovery algorithm on this resimulated data set, and comparing its output model against the model that was used to generate the resimulated data. Resimulation differs importantly from techniques such as Markov chain Monte Carlo (MCMC) sampling because the resimulated data is generated from a known structural model, providing a gold standard for calculating structure learning accuracy.

We introduce a new resimulation method called *hybrid resimulation (Hsim)*. Hybrid resimulation is a generalization of standard resimulation, allowing for only portions of the original data set to be resimulated. Standard resimulation is a special case of hybrid resimulation, in which all variables are resimulated. For this reason, this paper also refers to standard resimulation as *full resimulation (Fsim)*. After introducing these methods, we use a simulation study and a case study to compare the performance of Hsim and Fsim. The simulation study also shows that this is a very difficult task, and that standard resimulation does not provide strong evidence of an algorithm's performance on a data set.

There are many kinds of causal graphs, but this paper focuses on directed acyclic graphs (DAGs)[1,4] instantiated as structural equation models (SEMs)[5]. In the models considered in this paper, each variable's value is computed as a

sum of functions of its parents plus an independent error term unique to each variable. Let X and Y represent variables in a directed graph, and let V_X and V_Y represent the values that X and Y take at an arbitrary data point.[1]

$$V_X = \sum_{Y \in parents(X)} f_{YX}(V_Y) + \epsilon_X \quad (1)$$

ϵ is taken to be Gaussian in all cases, but both linear and nonlinear f functions are considered. An instantiated model (IM) is a fully specified SEM, with all functions and parameters specified with values.

This paper makes use of a single causal discovery algorithm, the Fast Greedy Equivalence Search (FGES) procedure[6]. At any point where we learn a causal graph from data, we are applying FGES to that data. It is typical for resimulation to reuse the same discovery algorithm at each stage.

Causal Discovery Performance Measures

Performance measures (PMs) summarize the closeness of a learned graph to a data generating graph. We focus on structural accuracy, which measures closeness of the graphical structures, and on the structural accuracy of learning directed acyclic graphs (DAGs) in particular. Different ways to measure closeness include numeric scores such as the F-measure[7] and Structural Hamming Distance[8]. To obtain a more detailed view of how well resimulation estimates PMs, this paper considers four standard measures of structural accuracy. These measures occur along 2 dimensions: the graphical features being compared, {adjacencies, edge orientations}, and the types of errors, {recall, precision}. An *adjacency* is a pair of nodes in a graph that are connected by an edge of any type. An *edge orientation* is the type and direction of an edge connecting two nodes. The specific error measures investigated in this paper are listed below. For brevity, let G_t be the true data generating graph, let G_l be the learned graph, let $\mathbf{A}$ be the set of adjacencies that are shared by G_t and G_l, and let $\mathbf{O}$ be the set of edge orientations that are shared by G_t and G_l.

1. Adjacency Recall (AR): $|\mathbf{A}|$ / the number of adjacencies in G_t.
2. Adjacency Precision (AP): $|\mathbf{A}|$ / the number of adjacencies in G_l.
3. Orientation Recall (OR): $|\mathbf{O}|$ / the number of edge orientations in G_t.
4. Orientation Precision (OP): $|\mathbf{O}|$ / the number of edge orientations in G_l.

Full Resimulation (Fsim)

Full Resimulation (Fsim) error estimation uses the following workflow: (input) data $\rightarrow$ (structure search) DAG $\rightarrow$ (parameterize the DAG) IM $\rightarrow$ (full simulation) data $\rightarrow$ (structure search) DAG $\rightarrow$ compare full DAGs. Each step is described in this section.

First, data are required as input. In our simulation study this data is sampled from a randomly generated DAG, and in our case study this data is provided by medical researchers. We use FGES[6] to learn a DAG, G1, from data, but other methods can be used instead. An instantiated model, IM1, is learned by fitting G1 to the data. IM1 determines a fixed joint distribution over the original data set's variables. In this paper, IM1 is always a linear Gaussian SEM; non-linear functions are only considered for generating the input data in our simulation study.

Samples are drawn from the distribution determined by IM1 until the number of simulated samples is equal to the number of samples in the original data set, resulting in the data set *Fdata*. DAG G2 is learned from this newly simulated data set. Finally, AR, AP, OR, and AP are calculated, with G1 as the gold standard graph and G2 as the estimate. G2 is dependent on random factors in the creation of Fdata from IM1, so estimation variance is reduced by repeating the last few steps of this process: drawing a new Fdata from IM1, learning a new G2 from the new Fdata, and comparing the new G2 to G1. This process can have arbitrarily many iterations, and the resulting PMs can be averaged. These averages are then used as the estimates of the true PMs.

[1]The distinction between variables and their values is necessary due to functions like $parents$ whose domain and range are in the space of variables, as opposed to the space of variable values.

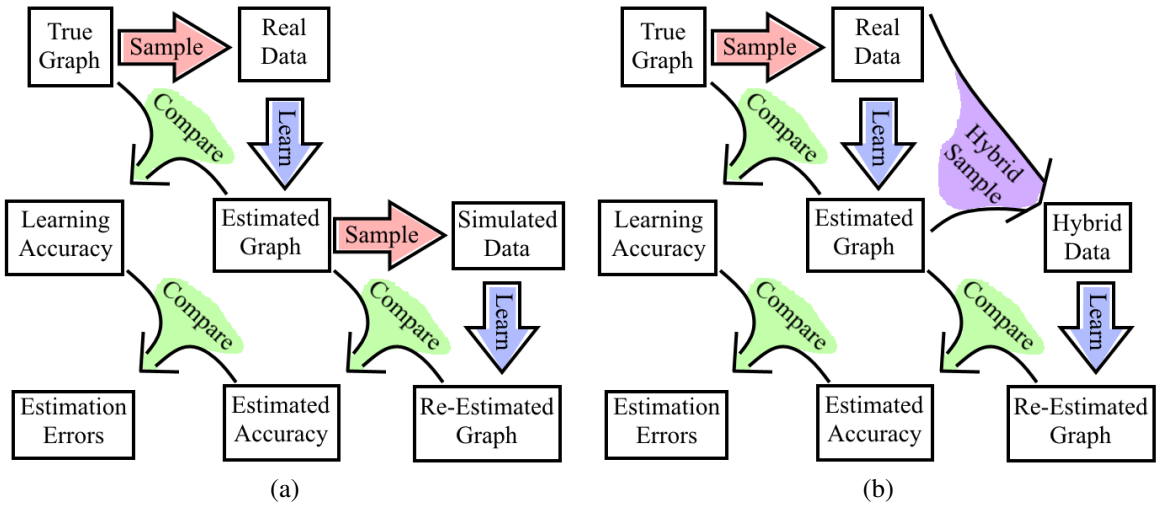


Figure 1: Diagrams describing the processes for calculating the estimation errors for (a) Full Resimulation (Fsim) and (b) Hybrid Resimulation (Hsim). These processes were iterated 500 times for the simulation study.

(Figure 1a) contains a diagram of how full resimulation is used to estimate accuracy, and how the errors of those estimates are calculated. (Figure 1b) contains a diagram of how hybrid resimulation is used to estimate accuracy, and how the errors of those estimates are calculated. The hybrid resimulation process is described in the next section.

Hybrid Resimulation (Hsim)

The high level workflow for Hybrid Resimulation (Hsim) Error Estimation is: (input) data $\rightarrow$ (structure search) DAG $\rightarrow$ (parameterize the DAG) IM $\rightarrow$ (hybrid simulation) data $\rightarrow$ (structure search) DAG $\rightarrow$ compare DAGs on specific edges. This process is described in this section.

Hsim is the same as Fsim until after IM1 is learned. Unlike Fsim, Hsim only uses IM1 to resimulate part of the data set. Specifically, it resimulates some variables and leaves other variables unchanged, i.e. *unresimulated*. The resimulated variables have their values drawn, row by row, from the conditional distribution determined by IM1 and the values of the unresimulated variables. For example, let R, X, and Y be all the variables, with only R being resimulated. Let V_R^i be the value of R at row i. To generate Hdata, V_R^i is sampled from $P_{IM1}(V_R|V_X = V_X^i, V_Y = V_Y^i)$.

Once Hdata is created, a DAG G2 is learned from it. Rather than comparing all edges in G1 to those in G2, AR, AP, OR, and OP are calculated from subgraphs of G1 and G2. Let G1H be the subgraph of G1 consisting only of edges oriented towards resimulated variables. Let G2H be the same, but with respect to G2. The PM estimates are calculated by treating G1H as the target graph estimated by G2H. As with Fsim, this process can be repeated multiple times to average out random factors in the procedure. For Hsim, the repeated process begins by randomly selecting different variables for resimulation. After the process is repeated as many times as is desired, average accuracy estimates are calculated, and used as the final estimates.

Simulation Study

We use a simulation study to compare the performance of Fsim and Hsim on two types of models: linear acyclic SEMs and nonlinear acyclic SEMs. This study also demonstrates the difficulty of this learning task, as Fsim performs worse than some readers might expect. (Figure 1) illustrates the high level process of this simulation study. We used an existing tool in the TETRAD package[2] to randomly generate each True Graph, instantiate SEMs from them, and generate Real Data from those SEMs[9]. These two cases were chosen because FGES was developed to work on linear SEMs, but not on nonlinear SEMs. Additionally, both Fsim and Hsim use learned linear models in their resimulation phases, which will naturally not accurately reproduce nonlinear causal relationships. The two cases are

[2]See http://www.phil.cmu.edu/tetrad/.

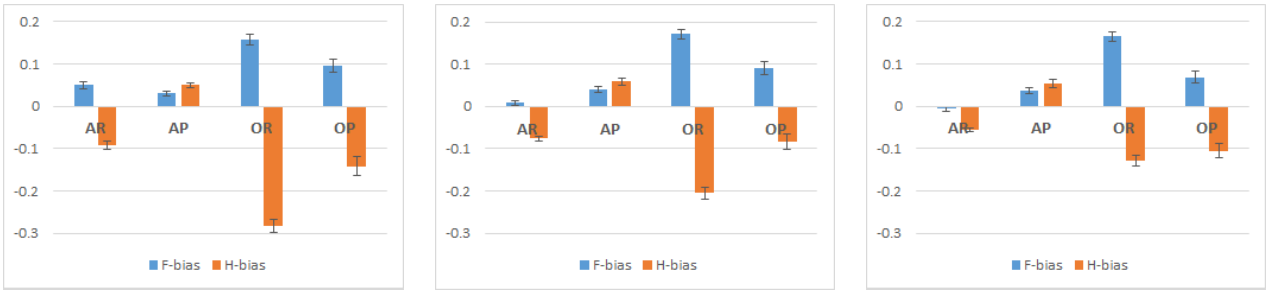
intended to differentiate resimulation accuracy estimates that stem from the following two differences: algorithm learning performance based on data that satisfy the algorithm's assumptions about how the data were generated and data that do not. Hybrid simulation aims to capture features of the data that might exist in the real data, but not in the fully resimulated data.

For this study, Fsim and Hsim used an iteration parameter of 100, meaning that their accuracy estimates for each of the simulated data sets were computed by averaging their accuracy estimates over 100 runs. Hybrid simulation can theoretically resimulate anywhere from no variables to all variables, although the end cases degenerate into all real data at one extreme and full resimulation at the other. For this study, Hsim was run with a resimulation size of 1 variable, selected randomly each time Hsim is run. This was chosen because it is the version of Hsim that is most different from Fsim, while still generating a hybrid of real and resimulated data.

All generated True Graphs are DAGs with 20 variables and 20 edges. The in-degrees and out-degrees of the nodes, as well as the graph's connectedness, were not restricted. Real Data were generated at sample sizes of 100, 300, and 900. The graphs were instantiated into full SEMs as either linear Gaussian models or nonlinear Gaussian models.

For the linear Gaussian models, we restrict (equation 1) such that $f_X Y(Y) = \beta_X YY$, with the edge parameter β being a constant drawn uniformly between 0.5 and 1.5. The Gaussian noise terms have mean 0 and variance drawn uniformly between 1 and 3. For each sample size, 500 different graphs were randomly generated and instantiated into models with randomly drawn parameter values. Each instantiated model was then used to generate a single data set, resulting in 500 sets of Real Data. FGES was run on each of these data sets to Learn an Estimated Graph, and Learning Accuracy was calculated by Comparing the Estimated Graph to the True Graph. Fsim and Hsim were then used to estimate those PMs, resulting in Estimated Accuracy values. Estimation Errors were calculated as the Estimated Accuracy measures minus the Learning Accuracy measures.

(Figure 2) shows the mean estimation errors for all four PMs when data are generated from linear models at sample sizes 100, 300, and 900. A value of 0 means the method is unbiased, positive values mean the method is biased towards overestimating the PMs, and negative values mean the method is biased towards underestimating the PMs.



(100) (300) (900)

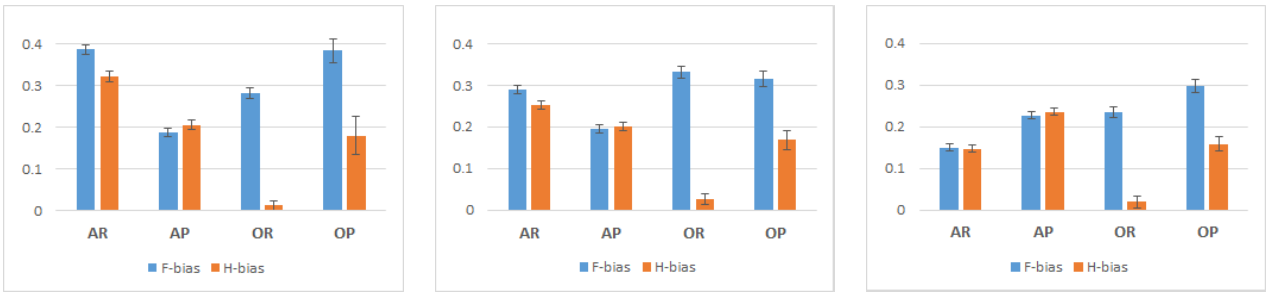
Figure 2: Simulation study results for linear models, showing mean estimation errors for AR, AP, OR, and OP at sample sizes 100, 300, and 900. Error bars represent 95% confidence intervals of the mean estimates shown.

Hsim and Fsim both had difficulty estimating orientation measures, with similar absolute errors. Fsim did well on recall measures, with average errors close to 0 in some cases. Hsim also did well on AR and AP, with absolute errors smaller than 0.1. An unexpected outcome of this simulation study is the variation in the signs of the errors made by Fsim and Hsim. Hsim's errors for AR, OR, and OP are consistently negative (underestimating performance), while Hsim's errors for AP are positive, although small. In contrast, Fsim's errors are positive (overestimating performance) or approximately 0. At the time of writing this paper it is unknown why these sign differences occur.

For these linear models, Fsim has preferable absolute errors by a small margin on most PMs at most sample sizes. Absolute errors appear to decrease for both Fsim and Hsim as sample size increases, with some exceptions. The Fsim OR error measure does not appear to decrease, while it does for Hsim. The converse seems to hold for the OP error measure. Interestingly, Fsim's OR has less absolute value than Hsim's OR on 100 samples, but by 900 samples Hsim's absolute OR error has become slightly less than Fsim's. Hsim is more conservative overall, tending to underestimate

rather than overestimate PMs, which in some cases might be preferable.

We choose the nonlinear Gaussian model functions and parameters so that the variables in the data set would fail a White test[10], but the correlation matrix of the data does not contain extreme values close to or equal to 1. A sine component was added to each edge, so that node X's value V_X is calculated as $V_X = [\sum_{Y \in parents(X)} \alpha_{YX} V_Y + \beta_{YX} sin(V_Y / \gamma_{YX})] + \epsilon_X$. α parameters were chosen uniformly between 0.5 and 1.0, β was chosen between 5 and 8, and γ was chosen between 1.0 and 1.5. ϵ is a Gaussian noise term with mean 0 and variance 1. (Figure 3) shows the results for the nonlinear simulations.



(100) (300) (900)

Figure 3: Simulation study results for nonlinear models, showing estimation errors for AR, AP, OR, and OP at sample sizes 100, 300, and 900. Error bars represent 95% confidence intervals of the mean estimates shown.

Unlike in the linear case, both Fsim and Hsim tended to overestimate PMs in the nonlinear case, with all of the average errors being positive. Compared to the linear case, overall errors were dramatically worse, with average estimation errors soaring to almost 0.4 for some PMs, an enormous amount of error considering the PMs are bounded between 0.0 and 1.0. Fsim and Hsim performed equally terribly for the recall measures, however Hsim performed much better at the orientation measures. Somewhat surprisingly, Hsim achieves an almost ideal error for OR in all three sample sizes, performing much better on that measure for the nonlinear case than it did for the linear case. As in the linear case, accuracy estimation generally improves for both Fsim and Hsim as sample size increases. There was little difference between Hsim and Fsim on the standard deviation of accuracy estimate errors (not shown), which for AR, AP, and OR ranged from 0.05 to 0.15 for all cases and methods, while for OP they typically ranged from 0.15 to 0.25.

Case Study

Fsim and Hsim were both used to estimate FGES's accuracy when applied to the Breast Cancer Wisconsin (Diagnostic) Data Set[11]. ID number and diagnosis variables were both removed, leaving 30 real-valued (continuous) variables with 569 samples. Fsim and Hsim were run in the same way that they were in the simulation study: FGES was used as the sole causal discovery method for all steps where graphs are learned from data, both methods were repeated 100 times, their results were averaged to produce the final accuracy estimates, and Hsim was run with a resimulation size of 1 variable. The results of both methods are provided in (Table 1).

Table 1: Fsim and Hsim accuracy estimates for FGES applied to the Breast Cancer Wisconsin (Diagnostic) Data Set.

Method	AR	AP	OR	OP
Fsim	0.69	0.77	0.56	0.64
Hsim	0.78	0.92	0.54	0.62

Compared to Fsim, Hsim estimates better AR, much better AP, and similar OR and OP. It is interesting to note that the average estimation errors made by Fsim and Hsim for AP on both the linear and nonlinear simulation studies were very similar for all sample sizes, never differing by more than 0.02. In contrast, the AP estimates for this real data set differ by almost 0.15, showing that Fsim and Hsim can produce very different estimates for individual data sets.

Discussion

This paper introduces a new method for estimating the accuracy of causal discovery methods—hybrid resimulation—and compares it to standard resimulation. The learning task is very difficult, and so neither hybrid or standard resimulation produce ideal results. Each performs well at some tasks under some circumstances. Overall, hybrid resimulation outperformed standard resimulation in the nonlinear simulations, and in some parts of the linear simulations.

The simulation study done in this paper could be enriched in a variety of ways. Resimulation sizes other than 1 variable could be used for Hsim to see how performance changes. The number of variables and edges in the true graphs could be varied as well. Some measured variables could be made latent, to see whether the resimulation methods are able to detect the errors introduced by latent confounders. Algorithms other than FGES could be used.

Using algorithms that can learn nonlinear relationships may improve estimates of causal discovery performance on nonlinear data. Since performance varied significantly between Fsim and Hsim, it might also be fruitful to consider schemes that aggregate the results of different kinds of resimulation to produce a more reliable accuracy estimate. For example, Fsim and Hsim often give different signs to their errors, so one could compute both and use them as bounds for the actual errors.

Acknowledgements

The authors would like to thank Bryan Andrews for assistance with generating the nonlinear data, and the UCI Machine Learning Repository for hosting the Breast Cancer Wisconsin (Diagnostic) Data Set. Research reported in this publication was supported by grant U54HG008540 awarded by the National Human Genome Research Institute through funds provided by the trans-NIH Big Data to Knowledge (BD2K) initiative. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health. This project was also funded, in part, by grant #4100070287 from the Pennsylvania Department of Health. The Department specifically disclaims responsibility for any analyses, interpretations, or conclusions.

References

1. Spirtes P, Glymour CN, Scheines R. Causation, prediction, and search. MIT press; 2000.
2. Tsamardinos I, Brown LE, Aliferis CF. The max-min hill-climbing Bayesian network structure learning algorithm. Machine learning. 2006 Oct 1;65(1):31-78.
3. Aliferis CF, Statnikov A, Tsamardinos I, Mani S, Koutsoukos XD. Local causal and markov blanket induction for causal discovery and feature selection for classification part i: Algorithms and empirical evaluation. Journal of Machine Learning Research. 2010;11(Jan):171-234.
4. Pearl J. Causality. Cambridge university press; 2009 Sep 14.
5. Bollen KA. Structural equations with latent variables. Wiley & Sons; 1989.
6. Ramsey J, Glymour M, Sanchez-Romero R, Glymour C. A million variables and more: the Fast Greedy Equivalence Search algorithm for learning high-dimensional graphical causal models, with an application to functional magnetic resonance images. International journal of data science and analytics. 2017 Mar 1;3(2):121-9.
7. Powers DM. Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation.
8. de Jongh M, Druzdzel MJ. A comparison of structural distance measures for causal Bayesian network models. Recent Advances in Intelligent Information Systems, Challenging Problems of Science, Computer Science series. 2009 Jan 1:443-56.
9. Ramsey JD, Malinsky D. Comparing the Performance of Graphical Structure Learning Algorithms with TETRAD. arXiv preprint arXiv:1607.08110. 2016 Jul 27.
10. White H. A heteroskedasticity-consistent covariance matrix estimator and a direct test for heteroskedasticity. Econometrica: Journal of the Econometric Society. 1980 May 1:817-38.
11. Street WN, Wolberg WH, Mangasarian OL. Nuclear feature extraction for breast tumor diagnosis.