Learning Causal Models That Make Correct Manipulation Predictions With Time Series Data

Mark Voortman VOORTMAN@SIS.PITT.EDU
Decision Systems Laboratory
School of Information Sciences
University of Pittsburgh
Pittsburgh, PA, 15260, USA

Denver Dash DENVER.H.DASH@INTEL.COM
Intel Research
Pittsburgh, PA, 15213, USA

Marek J. Druzdzel MAREK@SIS.PITT.EDU
Decision Systems Laboratory
School of Information Sciences
University of Pittsburgh
Pittsburgh, PA, 15260, USA

Editor: Isabelle Guyon, Dominik Janzing and Bernhard Schölkopf.

Abstract

One of the fundamental purposes of causal models is using them to predict the effects of manipulating various components of a system. It has been argued by Dash (2005, 2003) that the *Do* operator will fail when applied to an equilibrium model, unless the underlying dynamic system obeys what he calls *Equilibration-Manipulation Commutability*. Unfortunately, this fact renders most existing causal discovery algorithms unreliable for reasoning about manipulations. Motivated by this caveat, in this paper we present a novel approach to causal discovery of dynamic models from time series. The approach uses a representation of dynamic causal models motivated by Iwasaki and Simon (1994), which asserts that all “causation across time" occurs because a variable's derivative has been affected instantaneously. We present an algorithm that exploits this representation within a constraint-based learning framework by numerically calculating derivatives and learning instantaneous relationships. We argue that due to numerical errors in higher order derivatives, care must be taken when learning causal structure, but we show that the Iwasaki-Simon representation reduces the search space considerably, allowing us to forego calculating many high-order derivatives. In order for our algorithm to discover the dynamic model, it is necessary that the time-scale of the data is much finer than any temporal process of the system. Finally, we show that our approach can correctly recover the structure of a fairly complex dynamic system, and can predict the effect of manipulations accurately when a manipulation does not cause an instability. To our knowledge, this is the first causal discovery algorithm that has demonstrated that it can correctly predict the effects of manipulations for a system that does not obey the EMC condition.

Keywords: Causal discovery, dynamic systems, manipulations.

1. Introduction

One of the fundamental purposes of causal models is the prediction of the effects of manipulating various components of a system. It has been argued by Dash (2005, 2003) that the *Do* operator will fail when applied to an equilibrium model unless the underlying dynamic system obeys what he calls *Equilibration-Manipulation Commutability (EMC)*, a principle which is illustrated by the graph in Figure 1. In this figure, a dynamic system S, represented by a set of differential equations, is depicted on the upper-left. S has one or more equilibrium points such that, under the initial exogenous conditions, the equilibrium model $\tilde{S}$, represented by a set of equilibrium equations, will be obtained after sufficient time has passed. There are thus two approaches for making predictions of manipulations on S on time-scales sufficiently long for the equilibrations to occur. One could start with $\tilde{S}$ and apply the *Do* operator to predict manipulations. This is path A in Figure 1, and is the approach taken whenever a causal model is built from data drawn from a system in equilibrium. Alternatively, in path B the manipulations are performed on the original dynamic system which is then allowed to equilibrate; this is the path that the actual system takes. The EMC property is satisfied if and only if path A and path B lead to the same causal structure.

$$\begin{tikzcd}
S \arrow[rr, "\textit{Equilibration}"] \arrow[d, "\textit{Manipulation}"] \arrow[rrd, dotted, bend left=60, "A"] \arrow[rd, dotted, bend right=60, "B"'] & & \tilde{S} \arrow[d, "\textit{Manipulation}"] \\
\hat{S} \arrow[r, "\textit{Equilibration}"] & \tilde{\hat{S}} \arrow[r, phantom, "\overset{?}{=}"] & \hat{\tilde{S}}
\end{tikzcd}$$

Figure 1: Equilibration-Manipulation Commutability provides sufficient conditions for an equilibrium causal graph to correctly predict the effect of manipulations.

As an example of a system that obeys the EMC condition, consider a body of mass m dangling from a damped spring. The mass will stretch the spring to some equilibrium position $x = mg/k$ where k is the spring constant. As we vary m and allow the system to come to equilibrium, the value of x gets affected according to this relation. The equilibrium causal model $\tilde{S}$ of this system is simply $m \to x$. If one were to manipulate the spring directly and stretch it to some displacement $x = \hat{x}$, then the mass would be independent of the displacement, and the correct causal model is obtained by applying the *Do* operator to this equilibrium model.

Alternatively, one could have started with the original system S of differential equations of the damped simple-harmonic oscillator by explicitely modeling the acceleration $a = mg - kx - \alpha v$, where α is the dampening constant, and the velocity v. S can likewise be used to model the manipulation of x by applying the *Do* operator to a, v, and x simultaneously, ultimately giving the same structure as was obtained by starting with the equilibrium model. For examples of systems that do not obey the EMC condition, we refer the reader to Dash (2005, 2003) and the model shown later in this paper.

Unfortunately, requiring a system to obey the EMC condition renders most existing causal discovery algorithms unreliable for reasoning about manipulations, unless the details of the underlying dynamics of the system are explicitly represented in the model. Most classical causal discovery algorithms in AI make use of the class of independence constraints found in the data to infer causality between variables, assuming the faithfulness assumption (e.g., Spirtes et al., 2000; Pearl and Verma, 1991; Cooper and Herskovits, 1992). These methods will not be guaranteed to obey EMC if the observation time-scale of the data is long enough for some process in the underlying dynamic system to go through equilibrium. On the other hand, there have been previous approaches for learning dynamic causal models and Bayesian networks. Friedman et al. (1998) learn the structure of first-order Markov model by using time series data, and it would be straightforward to extend these approaches to higher-order Markovian models. However, the search space rapidly gets very large when searching for arbitrary dependencies across time.

Our approach, by contrast, uses an alternative representation of a dynamic system, explicitely modeling derivatives (or differences) of variables. It is beyond the scope of this paper to perform a quantitative comparison of prediction to these other approaches, however, we argue here that the representation that we learn helps us constrain the search space, and we expect that this reduction in complexity will make our algorithm perform better in practice and be more efficient than methods that try to learn fixed-order Markov structures for all variables.

2. Representation and Assumptions

Our approach uses a representation of dynamic causal models inspired by Iwasaki and Simon (1994), which asserts that all “causation across time" occurs because a variable’s derivative has been affected instantaneously. Iwasaki and Simon called these models “mixed causal structures". We use a slightly modified version of them and we call them “differential-based dynamic causal models" (DBD causal models, for short).

We use the notation $X^{(n)}$ to denote the n-th order derivative (or discrete version thereof) of variable X, and we use the convention that $X^{(0)} = X$.

Definition 1 (DBD graphs) *Differential-based dynamic causal graphs over a set of time-dependent variables X are discrete-time directed acyclic causal graphs, in which all “change across time" of a variable X occurs because there exists some n such that $X^{(n)}$ is being caused contemporaneously. That is, an edge exists from variable $Y_t \to X_{t+1}$ only if $Y_t = X_t^{(1)}$ or $Y_t = X_t$, in which case the parent set of X_{t+1} is $\{X_t, X_t^{(1)}\}$.*

The reason we constrain the parent set of X_{t+1} to be $\{X_t, X_t^{(1)}\}$ when X_t^1 is determined, is simply that, by definition,

$$X_{t+1} = X_t + X_t^{(1)} dt.$$

DBD models are unique in that they focus on uncovering *contemporaneous* causal relations that impact *derivatives* of some variables. They are motivated by real physical systems based on classical mechanics. For example, systems governed by Newton’s 2nd Law are archetypical causal systems: some “force" acts on a body, “causing" it to accelerate. The acceleration of the object, in turn, causes it to change velocity, which can cause the object to change position. The DBD reprentation assumes that all causation can be described in terms of “forces" causing a variable to change by impacting a derivative of some order instantaneously.

We show an example DBD graph in Figure 2. We will also use this example to illustrate the algorithm in one of the next sections. In the graph, two kinds of arcs are used: *solid arcs* that denote instantaneous causation, and *dashed arcs* that denote causation across time. The dashed arcs were called *integration links* by Iwasaki and Simon (1994) because they always point from a derivative of order n to a derivative of order $n-1$. The variables that have derivatives in

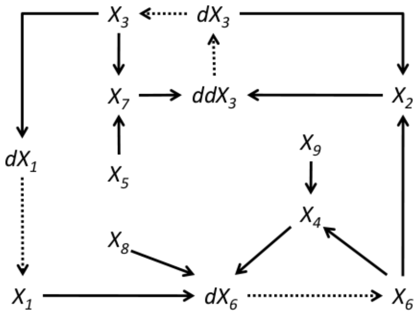

Figure 2: The DBD graph we used to simulate data.

the model are called *dynamic*, as they are solely responsible for the dynamics of the system. Assuming this representation, the learning algorithm has to find out which variables are dynamic and find the instanteneous causal arcs between variables and derivatives. It is important to note that our algorithm only learns contemporaneous causality, all dynamic behavior is then determined by integration over time.

In a dynamic structure, different causal equilibrium models may exist over different time-scales. Which equilibrium models will be obtained over time are determined by the time-scales at which variables equilibrate. The causal structures are derived from the equations by applying the causal ordering algorithm (Iwasaki and Simon, 1994) and by assuming that at fast time-scales, the slower moving variables are relatively constant. In the example of Figure 2, the time-scales could be such that $\tau_6 \ll \tau_3 \ll \tau_1$, where τ_i is the time-scale of variable X_i, in which case, at time $t \sim \tau_6$ it would be safe to assume that X_3 and X_1 are approximately constant. Under these time-scale assumptions, Figure 3 shows the different (approximate) models that exist for the graph in Figure 2.

One obvious approach to learning the graph of Figure 2 (assuming no derivative variables are present in the data), is to try to learn an arbitrary-order dynamic Bayesian network, for example using the method of Friedman et al. (1998). Figure 4 shows the second order Markov graph that a perfect DBN oracle would produce for this system. The problem with learning an arbitrary Markov model to represent this dynamic system is that there are no constraints as to which variables may affect other variables across time, so in principle, the search space could be unneccessarily large. The DBD representation, on the other hand, implies specific rules for when variables can affect other variables in the future (when they instantaneously effect some derivative of the variable). Given that a derivative $X^{(n)}$ is being instantaneously caused, DBDs also provide constraints on what variables can effect all $X^{(i)}$ for $i \neq n$.

We now state three conjectures concerning DBD models that are useful in explicating these constraints. Conjecture 1 is used to constrain the search space by limiting the number of possible dynamic variables. Conjecture 2 states that only one of the derivatives of a variable, or the variable itself, can have an incoming arc. Conjecture 3 is used to direct additional edges that are not oriented by the regular PC algorithm: If one of the derivatives of a variable has an incoming edge and the variable itself has an undirected edge, then the edge of the variable must be outgoing. This is necessary, otherwise it would conflict with Conjecture 2.

Conjecture 1 *Every non-exogenous root node that is present in the independence structure at time $t = 0$ is a dynamic variable.*

Conjecture 2 *Let A, B and C be different variables in a DBD model, and $\dot{A}$ any order derivative of A. If the model contains an arc $A \leftarrow B$, then it does not contain the arc $\dot{A} \leftarrow C$.*

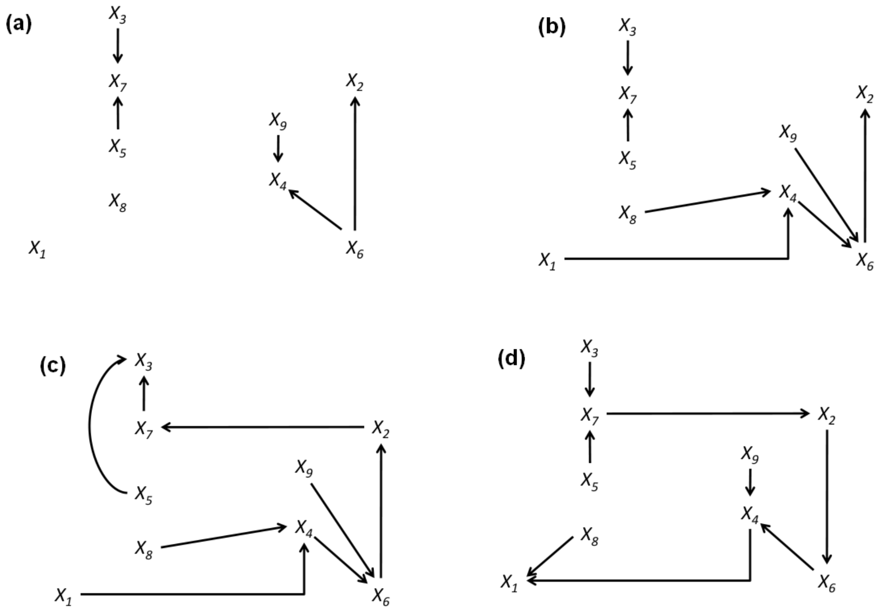

Figure 3: The different equilibrium models that exist in the sytem over time. (a) The independence constraints that hold when $t \sim 0$. (b) The independence constraints when $t \sim \tau_6$. (c) The independence constraints when $t \sim \tau_3$. (d) The independence constraints after all the variables are equilibrated, $t \gtrsim \tau_1$.

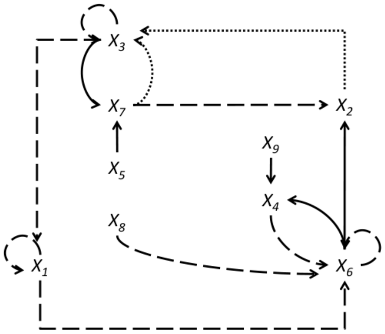

Figure 4: The second order Markov graph of the system. Thick dashed lines represent first-order Markov relations. Thin dotted lines represent second-order relations.

We use the standard notation $X - Y$ to indicate that either $X \rightarrow Y$ or $Y \rightarrow X$.

Conjecture 3 *Let A, B and C be different variables in a DBD model, and $\dot{A}$ any order derivative of A. If $A - B$ and $\dot{A} - C$, then the edge $A - B$ must be oriented $A \rightarrow B$.*

Our algorithm is based on the PC algorithm (Spirtes et al., 2000), although we could have used any other causal discovery algorithm as well. Besides the assumptions required for the PC algorithm, we make several additional assumptions. First, we assume that the system is stable, i.e., every dynamic variable must be part of a feedback loop. This implies that the highest order derivative of each dynamic variable must have at least one incoming arc. Second, exogenous variables are held constant over time and thus easily detectable in a data set. Third, in order for our algorithm to discover the dynamic model, it is necessary that the time-scale of the data is much finer than any temporal process of the system. This ensures that we are learning the dynamic model and not an equilibrium model. Finally, we assume that, apart from variable derivatives, the system is *causally sufficient* (i.e., there are no latent common causes).

3. The Algorithm

We present an algorithm that exploits the DBD representation within a constraint-based learning framework. The aim is to learn DBD models like the one given in Figure 2 directly. The input data[1] consisted of multiple time series that were generated first by parametrizing the model of Figure 2 with linear equations with independent Gaussian error terms, then by choosing different initial conditions for exogenous and dynamic variables and simulating 10000 discrete time steps. The integral equations have no noise, because they involve a deterministic relationship.

All derivatives of variables have been omitted from the data; thus, part of the challenge was that our method had to infer from the data which variables were changing due to the presence of derivatives and which were changing due to contemporaneous causation. Since calculating higher order derivatives using differences is sensitive to numerical errors, we opt for an incremental approach that gradually adds derivatives to the data set only when necessary, and exploits constraints given by our conjectures about feasible structures in these DBD graphs.

Our algorithm can be described in a few sentences: First we start with the original variables given in the data set and try to learn the instantaneous independence structure S_0^0 beteween non-derivative variables. This structure (plus our conjectures above) constrain which variables may be affected by derivatives. There may be multiple possible sets $S_0^1, S_1^1, \ldots, S_m^1$ of variables that could be consistent with S_0^0 and our conjectures. We then try to learn additional structure S_j^{i+1} with these new sets of variables assuming all links in S_j^i are correct. We recursively traverse the tree until we reach a set of maximum-order derivative models S_j^n, where n is an input into the algorithm. In instances where the structure from S_j^{i+1} contradicts structure from S_j^i, we assume S_j^i is correct. The output of the algorithm is then the complete set of consistent n-th order graphs.

To illustrate the algorithm, we will use the example model from Figure 2. To find out which variables are dynamic, we run the PC algorithm on a data set containing only non-derivative variables. The resulting structure will be the graph in Figure 3-a. The following four disconnected graphs will be discovered, and using Conjecture 1 we can find which variables are dynamic:

- X_1; this variable has to be dynamic, because it is not exogenous.
- X_8; this variable is exogenous and, therefore, not dynamic.
- $X_5 \rightarrow X_7 \leftarrow X_3$; X_5 is exogenous, X_7 is instanteneously caused and not dynamic, X_3 is not exogenous and, therefore, dynamic.
- $X_2 - X_6 \rightarrow X_4 \leftarrow X_9$; either X_2 or X_6 is dynamic, X_4 is not, and X_9 is exogenous.

Summarizing, X_1, X_3, and either X_2 or X_6 are dynamic variables so there are only two competing models.

[1] Downloadable from http://www.causality.inf.ethz.ch/repository.php?id=16

In the second step, for each of the competing models, the first order derivatives of the dynamic variables are added to the data set and the PC algorithm is executed again. The competing model in which X_2 is a dynamic variable will lead to an inconsistent structure, because there will be a v-structure into X_2, namely $dX_3 \rightarrow X_2 \leftarrow X_6$. This violates Conjecture 2 and so the structure is inconsistent. The other competing model is consistent, although no derivative of X_3 has an incoming edge. Therefore, as the last step, we add the second derivative of X_3 to the data set and run PC again to retrieve the original structure. We used Conjecture 3 as an extra rule to orient edges.

4. Prediction of Manipulations

The following results[2] were obtained by using the data and applying the instructions described in Appendix A. After running our algorithm on the data to obtain a causal structure, we estimated the coefficients in the equations in order to be able to make quantitative predictions. In the next step, we used the model and the values of the first four time steps in the data set to make predictions for time steps $\{5, 50, 100, 500, 1000, 2000, 4000, 10000\}$. We do not attempt to correct our predictions by using the data at times $t > 4$ when predicting later times, although doing this is possible and should improve our results.

The results are shown in Figure 5. Due to space constraints we chose not to present six tables, but instead calculated the average RMSE per time step for each manipulated variable. The graph shows that the error for the first few time steps is relatively small, but for all variables (except X_1) grows large in later times. Three variables in particular (X_2, X_7 and X_4) had astronomical errors in later times. These huge RMS errors are not indicative that our model was poor. In fact, in our case, since we generated the model, we could verify that the structure was exactly correct and the linear Gaussian parameters were very well identified. The reason for the unstable errors is that in the model of Figure 2, manipulating any variable except X_1 will approximately break the feedback loop of a dynamic variable and thus will in general result in an instability (Dash, 2003). Feedback variable X_1 is a relatively slow process, so breaking this feedback loop does not have a large effect on the feedback loops of X_3 and X_6. Thus our absolute rms error is expected to also be unstable all manipulations but X_1, simply because we are predicting such large values.

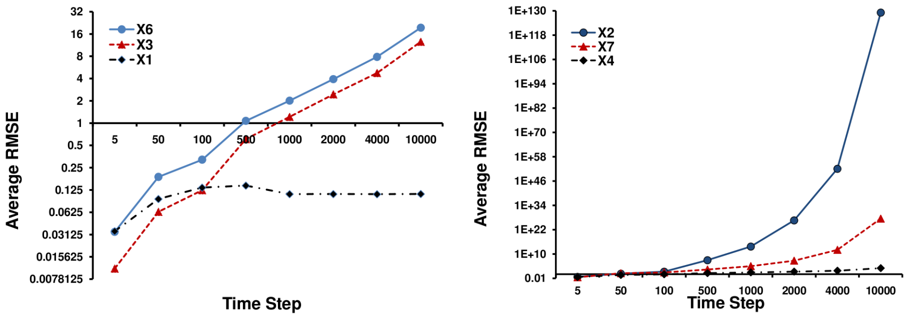

Figure 5: Average RMSE for each manipulated variable.

More important than getting the correct RMS error for these manipulations is the fact that our learned model correctly predicts that an instability will occur when any variable except X_1

[2] Results can be downloaded from http://pittsburgh.intel-research.net/~dhdash/causalitydata/rmse.zip.

is manipulated. In the absence of instability, our method has very low RMS error, as indicated by the curve of variable X_1 in Figure 5. This fact is significant, because the model retrieved from our system when variable X_1 is allowed to come to equilibrium will not obey the EMC condition (Dash, 2005). Thus, to our knowledge, we have presented the first algorithm that has demonstrated that it can correctly predict the effects of manipulations on systems that do not obey this condition.

5. Conclusions

We have described a first effort to construct an algorithm that can predict the effects of manipulations on systems that do not obey the EMC condition. We accomplish this by learning dynamic causal graphs in a representation very similar to that of Iwasaki and Simon. We have proposed a set of conjectures which are effective at constraining the search space for high-order Markovian relationships, which we expect will make this method more reliable and more efficient than other methods for learning temporal models, especially when higher-order relationships are present. We have shown that on a benchmark dataset generated from a fairly sophisticated dynamic system having multiple inter-related processes operating at widely varying time-scales, we were able to correctly learn the structure of the underlying system, and were able to predict that manipulating some variables in that system would result in an instability. Finally, in the absence of instabilities, we were able to predict with high accuracy the results of manipulating a variable, even far into the future. Future work will involve performing quantitative comparisons to other time-series methods. Also, although our conjectures formed useful heuristics for this method, we have been able to construct counter-examples where at least one of them is incorrect, so more work is needed to prove our existing conjectures and finding additional constraints on the search for derivatives.

Acknowledgments

This research was supported by the Air Force Office of Scientific Research grant FA9550–06–1–0243 and by Intel Research. All experimental data have been obtained using SMILE, a Bayesian inference engine developed at the Decision Systems Laboratory and available at http://genie.sis.pitt.edu/.

References

Gregory F. Cooper and Edward Herskovits. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9(4):309–347, 1992.

Denver Dash. *Caveats for Causal Reasoning*. PhD thesis, Intelligent Systems Program, University of Pittsburgh, Pittsburgh, PA, April 2003. http://etd.library.pitt.edu/ETD/available/etd-05072003-102145/.

Denver Dash. Restructuring dynamic causal systems in equilibrium. In Robert G. Cowell and Zoubin Ghahramani, editors, *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics (AIStats 2005)*, pages 81–88. Society for Artificial Intelligence and Statistics, 2005. (Available electronically at http://www.gatsby.ucl.ac.uk/aistats/).

Nir Friedman, Kevin Murphy, and Stuart Russell. Learning the structure of dynamic probabilistic networks. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI-98)*, pages 139–147. Morgan Kaufmann, 1998.

Yumi Iwasaki and Herbert A. Simon. Causality and model abstraction. *Artificial Intelligence*, 67(1):143–194, May 1994.

Judea Pearl and Thomas S. Verma. A theory of inferred causation. In J.A. Allen, R. Fikes, and E. Sandewall, editors, *KR–91, Principles of Knowledge Representation and Reasoning: Proceedings of the Second International Conference*, pages 441–452, Cambridge, MA, 1991. Morgan Kaufmann Publishers, Inc., San Mateo, CA.

Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, prediction, and search*. Springer Verlag, New York, NY, USA, second edition, 2000.

Appendix A. Pot-luck challenge: FACT SHEET .

Repository URL: http://www.causality.inf.ethz.ch/repository.php?id=16

Title: Mixed Dynamic Systems
Authors: Denver Dash, Mark Voortman, Marek Druzdzel
Contact name, address, email and website: Denver Dash, denver.h.dash@intel.com, http://pittsburgh.intel-research.net/~dhdash

Key facts:
Simulated time series data of 9 variables based on linear Gaussian models with no latent common causes, but with multiple dynamic processes at varying time-scales.
Training Data: 9 Variables, 10000 time series, each time series sampled at 12 distinct times (relative to when exogenous variables were first manipulated).
Testing Data: Manipulation data. Each of the 6 non-exogenous variables is manipulated and held fixed for the duration of the time series. This is repeated 100 times for each of the 6 variables.

Summary: A Mixed Dynamic System is one that consists of multiple dynamic processes operating at widely different time-scales. This data represents a 9 variable (labeled $X1 \ldots X9$) dynamic system with several dynamic processes acting on qualitatively different time scales from one another. The goal is to learn a causal model of the system with the training data, and then correctly predict the effects of various manipulations on the system (using the testing data for a quantitative measure of performance). This dataset was meant to be both simple and extremely challenging. All relations are linear with independent Gaussian error terms. There are no hidden confounders. However, we believe the inter-related dynamic processes will make prediction of manipulations challenging.

Training Data: The training data consists of 9 tab-separated text files (labelled X1.tsv, X2.tsv, etc.) one for each variable, and is arranged so that the rows in each file represent distinct time series for each variable (there are 10000 of these). That time series has been sampled at a few points in time after the exogenous variables of the system have been manipulated (all exogenous variables are held fixed for the duration of the time series). Specifically, the variables have been measured at the following discrete time intervals: t = {1, 2, 3, 4, 5, 50, 100, 500, 1000, 2000, 4000, 10000}, so there are 12 columns in each data file. Variables $X8$, $X5$ and $X9$ are all exogenous as can be verified by looking at X9.tsv, etc.

Test Data: The test data is organized into several (6x9 = 54) data files labeled Xi-manipj.tsv (For example X2-manip3.tsv shows the values of variable $X2$ when $X3$ has been manipulated and held fixed). Each variable in the set of endogenous variables {X1, X2, X3, X4, X6, X7} is manipulated 100 times for the entire 10000 time-step duration of each time series while the remaining variables are measured once at each of the 12 predetermined time-intervals. Thus each Xi-manipj.tsv file has 100 rows and 12 columns, and there are 9 files for each variable manipulated from the set {X1, X2, X3, X4, X6, X7}.

Evaluation: The objective of this problem is to use the first set of data labeled X*.tsv to build a model which is then able to predict the effects of manipulation on the system as given by the X*-manipN.tsv files. When predicting the effect of the manipulations, the goal is to predict the values of non-manipulated variables at times 5–10000 (columns 5 – 12) using the values of the previous times as input. For example, when predicting time 100 (column 7), you could use times 1, 2, 3, 4, 5, 50 (columns 1-6) as input. The output of the evaluation should be one table for each variable in the set {X1, X2, X3, X4, X6, X7} of manipulated variables. Each table should have 5 rows and 8 columns, one row for each variable in {X1, X2, X3, X4, X6, X7} $\setminus Xj$, (where Xj is the manipulated variable), and one column for each time in the set {5, 50, 100, 500, 1000, 2000, 4000, 10000}. The entry of the table is the RMS error (over the 100 runs) between the predicted value of the variable at that time and the actual value in the test data.