Journal of Machine Learning Research 1 (2000) 1-48 Submitted 4/00; Published 10/00

Causal learning without DAGs

David Duvenaud DUVENAUD@CS.UBC.CA
Daniel Eaton DEATON@CS.UBC.CA
Kevin Murphy MURPHYK@CS.UBC.CA
Mark Schmidt SCHMITDM@CS.UBC.CA
University of British Columbia
Department of Computer Science
2366 Main Mall
Vancouver, BC V6T 1Z4
Canada

Editor: TheEditor

Abstract

Causal learning methods are often evaluated in terms of their ability to discover a true underlying directed acyclic graph (DAG) structure. However, in general the true structure is unknown and may not be a DAG structure. We therefore consider evaluating causal learning methods in terms of predicting the effects of interventions on unseen test data. Given this task, we show that there exist a variety of approaches to modeling causality, generalizing DAG-based methods. Our experiments on synthetic and biological data indicate that some non-DAG models perform as well or better than DAG-based methods at causal prediction tasks.

Keywords: Bayesian Networks, Graphical models, Structure Learning, Causality, Interventions, Cell signalling networks, Bioinformatics.

1 Introduction

It is common to make causal models using directed acyclic graphs (DAGs). However, one problem with this approach is that it is very hard to assess whether the graph structure is correct or not. Even if we could observe "nature's graph", it probably would not be a DAG, and would contain many more variables than the ones we happened to have measured. Realistic mechanistic (causal) models of scientific phenomena are usually much more complex, involving coupled systems of stochastic partial differential equations, feedback, time-varying dynamics, and other complicating factors.

In this paper, we adopt a "black box" view of causal models. That is, we define causality in *functional* terms, rather than by committing to a particular representation. Our framework is as follows. Suppose we can measure d random variables, X_i, for $i = 1{:}d$. For example, these might represent the phosphorylation levels of different proteins. Also, suppose we can perform k different actions (interventions), A_j, for $j = 1:k$. For example, these might represent the application of different chemicals to the system. For simplicity, we will think of the actions as binary, $A_j \in \{0, 1\}$, where a value of 1 indicates that we performed action A_j. We *define* a causal model as one that can predict the effects of actions on the system, i.e., a conditional density model of the form $p(\mathbf{x}|\mathbf{a})$. These actions may or may not have been seen before, a point we discuss in more detail below. Note that our definition of causal model is even more general than the one given in Dawid (2009),

who defines a causal model as (roughly speaking) any model that makes conditional independence statements about the X and A variables; as Dawid points out, such assumptions may or may not be representable by a DAG.

To see that our definition is reasonable, note that it includes the standard approach to causality (at least of the non-counterfactual variety) as a special case. In the standard approach (see e.g., (Spirtes et al., 2000; Pearl, 2000; Lauritzen, 2000; Dawid, 2002)), we assume that there is one action variable for every measured variable. We further assume that $p(\mathbf{x}|\mathbf{a})$ can be modeled by a DAG, as follows:

$$p(X_1, \ldots, X_d | A_1 = 0, \ldots, A_d = 0, G, f) = \prod_{j=1}^{d} f_j(X_j, X_{\pi_j}) \tag{1}$$

where G is the DAG structure, π_j are the parents of j in G, and $f_j(X_j, X_{\pi_j}) = p(X_j|X_{\pi_j}, A_j = 0)$ is the conditional probability distribution (CPD) for node j, assuming that node j is not being intervened on (and hence $A_j = 0$). If node j *is* being intervened on, we modify the above equation to

$$p(X_1, \ldots, X_d | A_j = 1, A_{-j} = 0, G, f, g) = g_j(X_j, X_{\pi_j}) \prod_{k \neq j} f_k(X_k, X_{\pi_k}) \tag{2}$$

where $g_j(X_j, X_{\pi_j}) = p(X_j|X_{\pi_j}, A_j = 1)$ is the CPD for node j given that node j is being intervened on. In the standard model, we assume that the intervention sets the variable to a specific state, i.e., $g_j(X_j, X_{\pi_j}) = I(X_j = S_j)$, for some chosen target state S_j. This essentially cuts off the influence of the parents on the intervened-upon node. We call this the *perfect* intervention assumption. A real-world example of this might be a gene knockout, where we force X_j to turn off (so $S_j = 0$). The crucial assumption is that actions have local effects, and that the other f_j terms are unaffected.

If we do not know which variables an action affects, we can learn this; we call this the *uncertain* intervention model (Eaton and Murphy, 2007). In particular, this allows us to handle actions which affect multiple nodes. These are sometimes called "fat hand" actions; the term arises from thinking of an intervention as someone "sticking their hand" into the system, and trying to change one component, but accidently causing side effects. Of course, the notion of "fat hands" goes against the idea of local interventions. In the limiting case in which an action affects all the nodes, it is completely global. This could be used to model the effects of a lethal chemical that killed a cell, and hence turned all genes "off".

If we model $p(\mathbf{x}|\mathbf{a})$ by a DAG, and make the perfect intervention assumption, then we can make predictions about the effects of actions we have never seen before. To see this, suppose we have collected N samples from the non interventional regime, $\mathcal{D} = \{\mathbf{x}_n\}_{n=1}^N$, where $\mathbf{x}_n \sim p(\mathbf{x}|\mathbf{a} = \mathbf{0})$ (this is called observational data). We can use this data to learn the non-interventional CPDs f_j. Then we make a prediction about what would happen if we perform a novel action, say turning A_j on, by simply replacing f_j with g_j, which we assume is a delta function, $I(X_j = S_j)$. Of course, if the data is only observational, we will not, in general, be able to uniquely infer the DAG, due to problems with Markov equivalence. However, if some of the data is sampled under perfect interventions, then we can uniquely recover the DAG (Eberhardt et al., 2005, 2006).

The key question is: is the assumption of DAGs and perfect interventions *justified* in any given problem? What other models might we use? It seems that the only way to choose between methods in an objective way, without reference to the underlying mathematical representation, is to collect some real-world data from a system which we have perturbed in various ways, partition the data

into a training and test set, and then evaluate each model on its ability to predict the effects of interventions. This is what we do in this paper.

An important issue arises when we adopt this functional view of causality, which has to do with generalizing across actions. In the simplest case, we sample training data from regimes $p(\mathbf{x}|\mathbf{a}_1), \ldots, p(\mathbf{x}|\mathbf{a}_r)$, for r different action combinations, and then sample test data *from the same regimes*. We will see an example of this in Section 3.1, where we discuss the intracellular flow cytometry dataset analyzed in Sachs et al. (2005). In this setup, we sample data from the system when applying one chemical at a time, and then ask the model to predict the protein phosphorylation levels when the same chemical is applied.

A more interesting task is to assume that the test data is drawn from a different sampling regime than the training data. This clearly requires that one make assumptions about how the actions affect the variables. We will see an example of this in Section 3.2, where we discuss another flow cytometry dataset, used in the Dream 2008 competition. In this setup, we sample data from the system when applying one inhibitory chemical and one excitatory chemical at a time, but then ask the model to predict the protein phosphorylation levels when a novel pair of chemicals is applied. For example, we train on data sampled from $p(\mathbf{x}|a_1 = 1, a_2 = 1, a_3 = 0)$ and $p(\mathbf{x}|a_1 = 0, a_2 = 1, a_3 = 1)$, and test on data sampled from $p(\mathbf{x}|a_1 = 1, a_2 = 0, a_3 = 1)$. That is, we have seen A_1 and A_2 in combination, and A_2 and A_3 in combination, and now want to predict the effects of the A_1, A_3 combination. Another variation would be to train on data from $p(\mathbf{x}|a_1 = 1, a_2 = 0)$ and $p(\mathbf{x}|a_1 = 0, a_2 = 1)$, and test on data sampled from $p(\mathbf{x}|a_1 = 1, a_2 = 1)$. This is similar to predicting the effects of a double gene knockout given data on single knockouts.

The most challenging task is when the testing regime contains actions that were never tried before in the training regime, neither alone nor in combination with other actions. For example, suppose we train on data sampled from $p(\mathbf{x}|a_1 = 1, a_2 = 0)$ and test on data sampled from $p(\mathbf{x}|a_1 = 0, a_2 = 1)$. In general, these distributions may have nothing to do with each other. Generalizing to a new regime is like predicting the label of a novel word in a statistical language model. In general, this is impossible, unless we break the word down into its component pieces and/or describe it in terms of features (e.g., does it end in "ing", does it begin with a capital letter, what is the language of origin, what is the context that it was used in, etc). If we represent actions as "atomic", all we can do is either make the DAG plus perfect intervention assumption, or assume that the action has no affect, and "back-off" to the observational regime. We will compare these approaches below.

2 Methods

In this section, we discuss some methods for learning conditional density models to represent $p(\mathbf{x}|\mathbf{a})$, some based on graphs, others not. We will compare these methods experimentally in the next section. Code for reproducing these experiments will be made available at `www.cs.ubc.ca/~murphyk/causality`.

2.1 Approaches to Modeling Interventions

We consider several classes of methods for creating models of the form $p(\mathbf{x}|\mathbf{a})$:

1. **Ignore**: In this case, we simply ignore A and build a generative model of $P(X)$. This has the advantage that we gain statistical strength by pooling data across the actions, but has the disadvantage that we make the same prediction for all actions.

2. **Independent**: In this case, we fit a separate model $P(X|A)$ for each unique joint configuration of A. This is advantageous over the *ignore* model in that it makes different predictions for different actions, but the disadvantage of this model is that it does not leverage information gained between different action combinations, and can not make a prediction for an unseen configuration of A.

3. **Conditional**: In this case, we build a model of $P(X|A)$, where we use some parametric model relating the A's and X's. We give the details below. This will allow us to borrow strength across action regimes, and to handle novel actions.

2.2 Approaches based on DAGs

In the ignore case, we find the exact MAP DAG using the dynamic programming algorithm proposed in (Silander and Myllmaki, 2006) applied to all the data pooled together. We can use the same algorithm to fit independent DAGs for each action, by partitioning the data. In the conditional case, there are two ways to proceed. In the first case, which we call **perfect**, we assume that the interventions are perfect, and that the targets of intervention are known. In this case, it is simple to modify the standard BDeu score to handle the interventional data, as described in Cooper and Yoo (1999). These modified scores can then be used inside the same dynamic programming algorithm. In the second case, which we call **uncertain**, we learn the structure of an augmented DAG containing A and X nodes, subject to the constraint that there are no $A \rightarrow A$ edges or $X \rightarrow A$ edges. It is simple to modify the DP algorithm to handle this; see (Eaton and Murphy, 2007) for details.

2.3 Approaches based on undirected graphs

DAG structure learning is computationally expensive due to the need to search in a discrete space of graphs. In particular, the exact dynamic programming algorithm mentioned above takes time which is exponential in the number of nodes. Recently, computationally efficient methods for learning undirected graphical model (UGM) structures, based on L1 regularization and convex optimization, have become popular, both for Gaussian graphical models (Meinshausen and Buhlmann, 2006; Friedman et al., 2007; Banerjee et al., 2008), and for Ising models (Wainwright et al., 2006; Lee et al., 2006). In the case of general discrete-state models, such as the ternary T-cell data, it is necessary to use a *group* L1 penalty, to ensure that all the parameters associated with each edge get "knocked out" together. Although still convex, this objective is much harder to optimize (see e.g., (Schmidt et al., 2008) and (Duchi et al., 2008) for some suitable algorithms). However, for the small problems considered in this paper, we found that using L2 regularization on a fully connected graph did just as well as L1 regularization, and was much faster. The strength of the L2 regularizer is chosen by cross validation.

To apply this technique in the *ignore* scenario, we construct a Markov random field, where we create factors for each X_i node and each $X_i - X_j$ edge. For the *independent* scenario, one such Markov random field is learned for each action combination in the training set. In the *interventional*

scenario, we construct a conditional random field, in which we additionally create factors for each $X_i - A_j$ edge, and for each X_i, X_j, A_k triple (this is similar to a chain graph; see (Lauritzen and Richardson, 2002) for a discussion.) Since it does not contain directed edges, it is harder to interpret from a causal perspective. Nevertheless, in Section 3.1, we show that the resulting model performs very well at the task of predicting the effects of interventions.

2.4 Other methods

There are of course many other methods for (conditional) density estimation. As a simple example of a non graph based approach, we considered mixtures of K multinomials. In the ignore case, we pool the data and fit a single model. In the independent case, we fit a separate model for each action combination. In the conditional case, we fit a mixture of independent logistic regressions:

$$p(\mathbf{x}|\mathbf{a}) = \sum_k p(z=k) \prod_{j=1}^{d} p(x_j|z=k,\mathbf{a}) \tag{3}$$

where $p(z=k)$ is a multinomial, and $p(x_k|\mathbf{a}, z=k)$ is multinomial logistic regression. This is similar to a mixture of experts model (Jordan and Jacobs, 1994).

2.5 Summary of methods

In summary, we have discussed 10 methods, as follows: 3 models (Mixture Model, UGM or DAG), times 3 types (ignore, independent, conditional), plus perfect intervention DAGs. We did not try independently trained DAGs, because it was substantially slower than other methods (using exact structure learning), so we only consider 9 methods in total.

3 Experimental results

In the introduction, we argued that, in the absence of a ground truth graph structure (which in general will never be available), the only way to assess the accuracy of a causal model is to see how well it can predict the effects of interventions on unseen test data. In particular, we assume we are given a training set of $(\mathbf{a}, \mathbf{x})$ pairs, we fit some kind of conditional density model $p(\mathbf{x}|\mathbf{a})$, and then assess its predictive performance on a different test set of $(\mathbf{a}, \mathbf{x})$ pairs.

3.1 T-cell data

Flow cytometry is a method for measuring the "status" of a large number of proteins (or other molecules) in a high throughput way. In an influential paper in Science in 2005, Sachs et al. used flow cytometry to collect a dataset of 5400 samples of 11 proteins which participate in a particular pathway in T-cells. They measured the protein phosphorylation levels under various experimental conditions. Specifically, they applied 6 different chemicals separately, and measured the status of the proteins; these chemicals were chosen because they target the state of individual proteins. They also measured the status in the unperturbed state (no added chemicals).[1] Sachs et al. then

1. This original version of the data is available as part of the 2008 Causality Challenge. See the CYTO dataset at `http://www.causality.inf.ethz.ch/repository.php`.

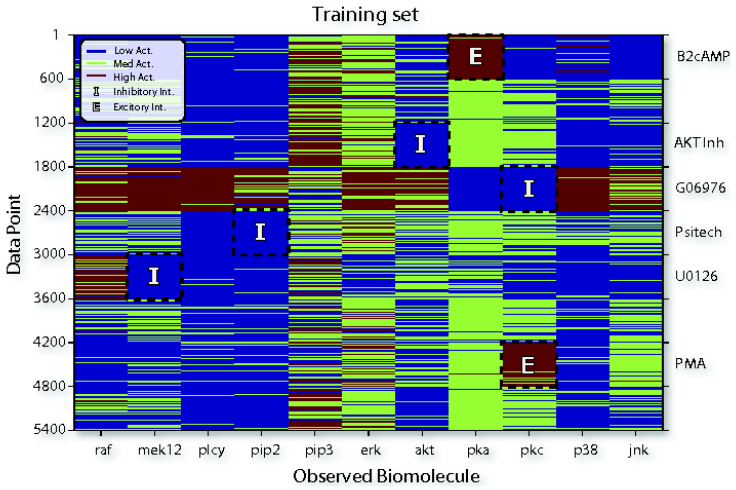

Figure 1: **T-cell data.** 3-state training data from (Sachs et al., 2005). Columns are the 11 measured proteins, rows are the 9 experimental conditions, 3 of which are "general stimulation" rather than specific interventions. The name of the chemical that was added in each case is shown on the right. The intended primary target is indicated by an E (for excitation) or I (for inhibition). There are 600 measurements per condition. This figure is best viewed in colour.

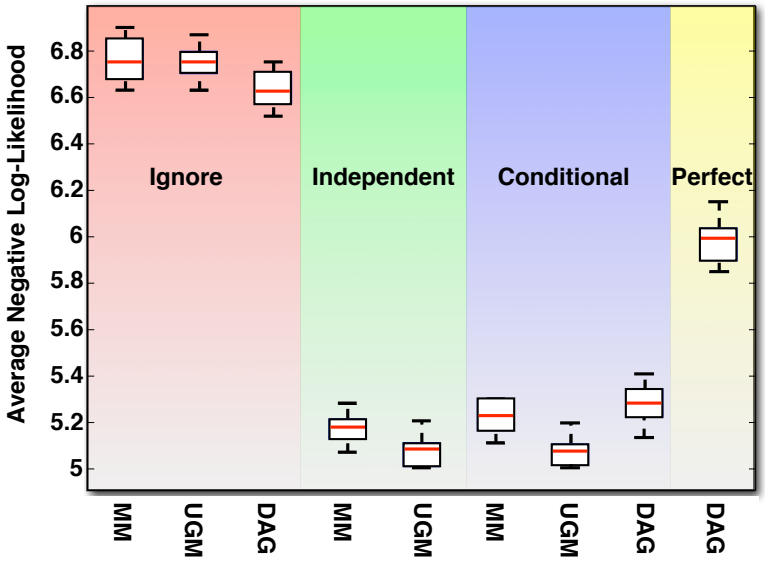

Figure 2: 10-fold cross-validated negative log likelihood on the T-cell data (lower is better). The methods are divided based on their approach to modeling interventions (*Ignore* the interventions, fit *Independent* models for each intervention, fit a *Conditional* model that conditions on the interventions, or assume *Perfect* interventions). Within each group, we sub-divide the methods into MM (mixture of multinomials), UGM (undirected graphical model), and DAG (directed acyclic graphical model).

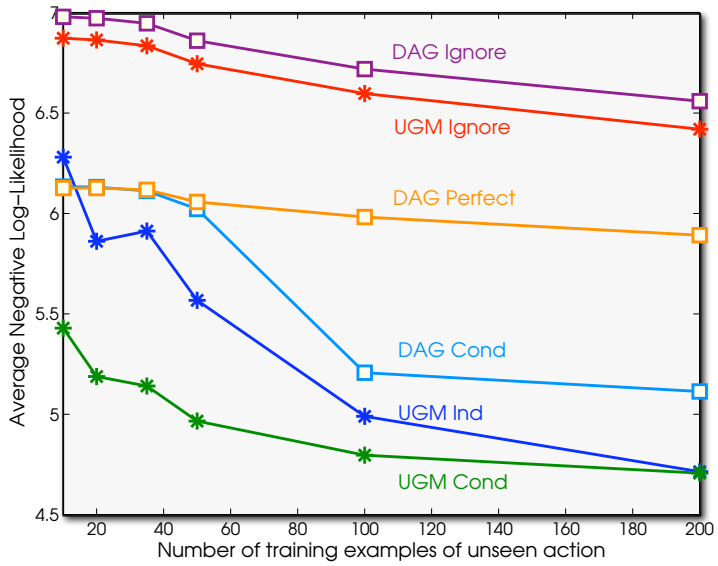

Figure 3: Average (per-case) negative log-likelihood on the T-cell test data as a function of the amount of training data for one particular action regime, given the data from all other action regimes. Results when choosing other actions for the "sparse training regime" are similar. "DAG Cond" is a DAG with uncertain interventions. "UGM Ind" is a UGM fit independently for each action.

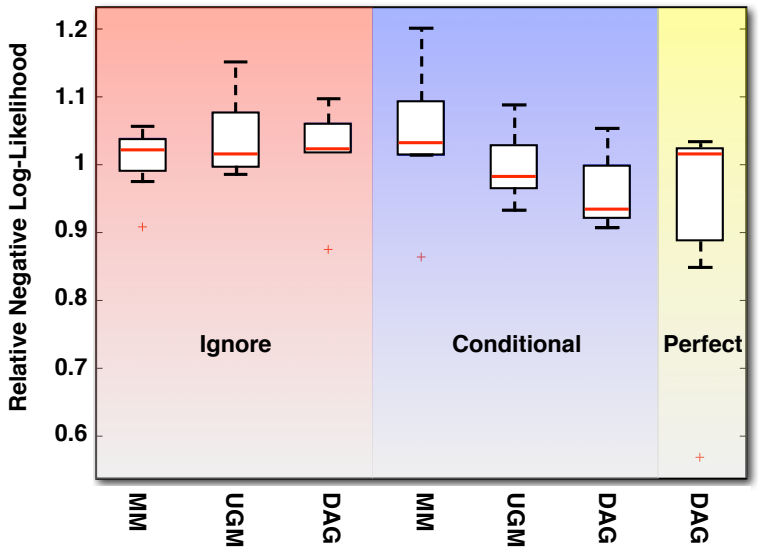

Figure 4: Negative log-likelihood on the T-cell data for different methods when predicting a novel action, using data from all the other actions as training. The boxplot shows the variation when different actions are chosen as the prediction targets. We plot performance relative to the best method for each chosen action, since some actions are easier to predict than others.

discretized the data into 3 states, representing low, medium and high activation (see Figure 1), and learned a DAG model using simulated annealing and the scoring function described in (Cooper and Yoo, 1999). The resulting DAG was quite accurate, in that it contained most of the known edges in the biological network, and few false positives. However, it is known that the "true" graph structure contains feedback loops, which cannot be modeled by a DAG. In addition, there are many variables in the "true" model that are not measured in the data. Hence assessing performance by looking at the graph structure is not ideal. Instead, we will measure predictive accuracy of the learned models.

We used the same *discretized* version of the data as in the original Sachs paper. There are 600 samples in each interventional regime, and 1800 samples in the observational regime, for a total of 5400 samples. There is no pre-specified train/test split in the T-cell data, so we have to make our own. A natural approach is to use cross validation, but a subtlety arises: the issue is whether the test set folds contain novel action combinations or not. If the test data contains an action setting that has never been seen before, in general we cannot hope to predict the outcome, since, for example, the distribution $p(\mathbf{x}|a_1 = 0, a_2 = 1)$ need have nothing in common with $p(\mathbf{x}|a_1 = 1, a_2 = 0)$.

Initially we sidestep this problem and follow the approach taken by Ellis and Wong (2008), whereby we assess predictive performance using 10-fold cross validation, where the folds are chosen such that each action occurs in the training and test set. Hence each training set has 540 samples and each validation set has 60 samples.

The results of evaluating various models in this way are shown in Figure 2. We see that the methods which ignore the actions, and pool the data into a single model, do poorly. This is not surprising in view of Figure 1, which indicates that the actions do have a substantial affect on the values of the measured variables. We also see that the approach that learns the targets of intervention (the *conditional DAG*) is significantly better than learning a DAG assuming that the interventions are perfect (see last two columns of Figure 2). Indeed, as discussed in Eaton and Murphy (2007), the structure learned by the uncertain DAG model indicates that each intervention affects not only its suspected target, but several of its neighbors as well. The better prediction performance of this model indicates that the perfect intervention assumption may not be appropriate for this data set. However, we also see that *all* the independent and conditional models not based on DAGs do as well or better than the DAG methods.

It was somewhat surprising how well the independent models did. This is presumably because we have so much data in each action regime, that it is easy to learn separate models. To investigate this, we considered a variant of the above problem in which we trained on all 600 samples for all but one of the actions, and for this remaining action we trained on a smaller number of samples (and tested only on this remaining action). This allows us to assess how well we can borrow statistical strength from the data-rich regimes to a data-poor regime. Figure 3 shows the results for several of the models on one of the actions (the others yielded largely similar results). We see that the conditional models need much less training data when faced with a novel action regime than independent models, because they can borrow statistical strength from the other regimes. Independent models need much more data to perform well. Note that even with a large number of samples, the perfect DAG model is not much better than fitting a separate model to each regime.

The logical extreme of the above experiment is when we get no training samples from the novel regime. That is, we have 600 training samples from each of the following: $p(\mathbf{x}|1, 0, 0, 0, 0, 0)$, $p(\mathbf{x}|0, 1, 0, 0, 0, 0)$, ... $p(\mathbf{x}|0, 0, 0, 0, 1, 0)$, and we test on 600 samples from $p(\mathbf{x}|0, 0, 0, 0, 0, 1)$, where the bit vector on the right hand side of the conditioning bar specifies the state of the 6 A_j action variables. We can then repeat this using leave-one-action out. The results are shown in

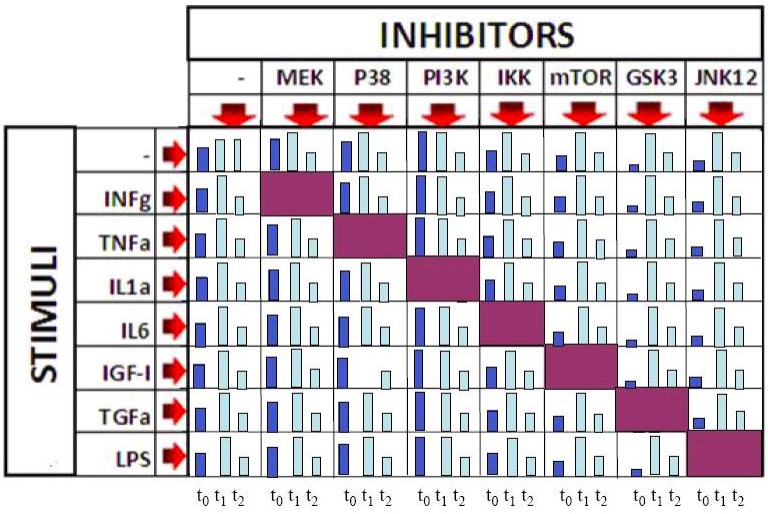

Figure 5: Dream 3 phosphoprotein data. See text for details.

Stimulus							Inhibitor								
INFg	TNFa	IL1a	IL6	IGF1	TGFa	LPS	MEK	P38	P13K	IKK	mTOR	GSK3	JNK12	X1	X17
0	0	0	0	0	0	0	0	0	0	0	0	0	0	5578	275
0	0	0	0	0	0	0	1	0	0	0	0	0	0	454	89
0	0	0	0	0	0	0	0	1	0	0	0	0	0	1001	99
						.								.	
						.								.	
0	0	0	0	0	0	1	0	0	0	0	0	1	0	22	33

Figure 6: The dream 3 training data represented as a design matrix. We treat each cell type and time point separately, and show the response of the 17 phosphoproteins to 58 different action combinations (58 is 8×8 minus the 6 test conditions shown in Figure 5.) Each 14-dimensional action vector has 0, 1 or 2 bits turned on at once. For example, the last row corresponds to stimulus=LPS, inhibitor = GSK3.

Figure 4. (We do not show results for the independently trained models, since their predictions on novel regimes will be based solely on their prior, which is essentially arbitrary.) We see that all methods do about the same in terms of predictive accuracy. In particular, the perfect DAG model, which is designed to predict the effects of novel actions, is actually slightly worse than conditional DAGs and conditional UGMs in terms of its median performance.

3.2 DREAM data

One weakness of the CYTO dataset discussed above is that the actions are only performed one at a time. A more recent dataset has been collected which measures the status of proteins under different action combinations.[2] This data is part of the DREAM 3 competition, which took place in November 2008. (DREAM stands for "Dialogue for Reverse Engineering and Assessment of Methods".) The data consists of measurements (again obtained by flow cytometry) of 17 phosphoproteins and 20 cytokines at 3 time points in 2 cell types under various combinations of chemicals (7 stimuli and

2. This data is available from `http://wiki.c2b2.columbia.edu/dream/index.php/The_Signaling-Response_Prediction_Challenge._Description`.

Team	MSE
PMF	1483
Linear regression	1828
Team 102	3101
Team 106	3309
Team 302	11329

Figure 7: Mean squared error on the DREAM 3 dataset, using the training/test set supplied with the challenge. Also listed is the performance of the three other teams who competed in the challenge.

7 inhibitors). In the challenge, the response of the proteins under various stimulus/ inhibitor pairs is made available, and the task is to predict the response to novel stimulus/ inhibitor combinations. In this paper, we focus on the phosphoprotein data. The data is illustrated in Figure 5. Another way to view this data is shown in Figure 6.

The DREAM competition defines a train/test split, and evaluates methods in terms of their mean squared error for predicting the responses of each variable separately to 6 novel action combinations. In Table 7, we show the scores obtained by the 3 entrants to the competition in November 2008. The method used by these teams has not yet been disclosed, although the organizer of the Dream competition (Gustavo Stolovitzky) told us in a personal communication that they are not based on graphical models. We also show two approaches we tried. The first uses simple linear regression applied to the 14-dimensional binary action vector $\mathbf{a}$ to predict each response X_j (since the methods are evaluated in terms of mean squared-error, this is equivalent to using a conditional DAG model with linear-Gaussian CPDs) We see that this beats all the submitted entries by a large margin. However, the significance of this result is hard to assess, because there is only a single train/test split. We also tried probabilistic matrix factorization, using $K = 3$ latent dimensions. This is similar to SVD/PCA but can handle missing data (see Salakhutdinov and Mnih (2008) for details). This choice was inspired by the fact that the data matrix in Figure 5 looks similar to a collaborative filtering type problem, where the goal is to "fill in" holes in a matrix. We see that PMF does even better than linear regression, but again it is hard to assess the significance of this result. Hence in the next section, we will discuss a synthetic dataset inspired by the design of the DREAM competition.

3.3 Synthetic Data

Since the DREAM data uses population averaging rather than individual samples, it does not contain enough information to learn a model of the underlying system. Thus, we sought to validate some of the approaches discussed here on a synthetic data set. To this end, we generated synthetic data sets that simulate the DREAM training/testing regime (i.e., where we train on pairs of actions and test on novel pairs).

We sought to generate a data set that has a clearly defined notion of intervention, but that is not a DAG. To do this we simulated data from a discrete structural equation model (SEM) (see Pearl (2000)). In particular, we generated a data set where each variable X_j is updated based on

$$p(X_j = 1|\mathbf{x}_{\pi_j}, \boldsymbol{\theta}_j) = \sigma(w_{0j} + \mathbf{w}_j^T \mathbf{x}_{\pi_j}) \quad (4)$$

$$p(X_j = -1|\mathbf{x}_{\pi_j}, \boldsymbol{\theta}_j) = 1 - p(X_j = 1|\mathbf{x}_{\pi_j}, \boldsymbol{\theta}_j) \quad (5)$$

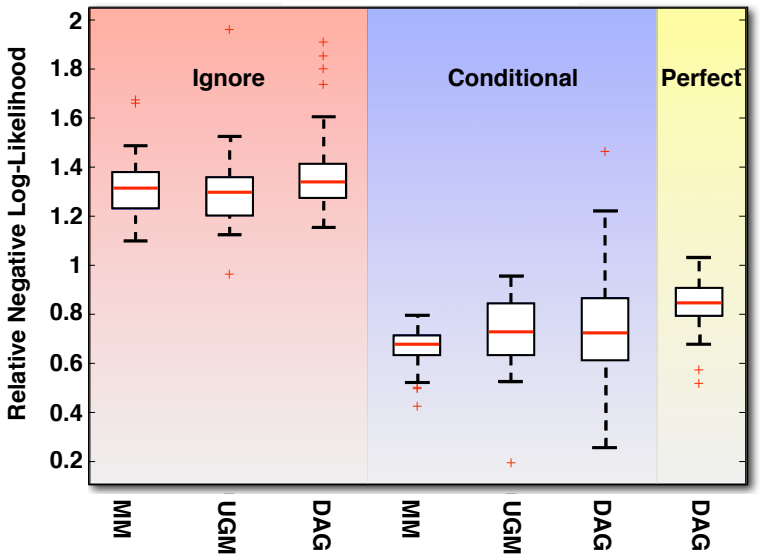

Figure 8: Negative log-likelihood for novel action combinations on synthetic data generated from a fully visible SEM. We plot NLL relative to the best performing method.

where $\sigma()$ is the sigmoid function $\sigma(x) \triangleq 1/(1+\exp(-x))$, and $\boldsymbol{\theta}_j = (w_{0j}, \mathbf{w}_j)$ are the paramters for each node; here w_{0j} is the bias term and $\mathbf{w}_j$ are the regression weights. We generated each w_0 from a standard Normal distribution, and to introduce strong dependencies between nodes we set each element of each $\mathbf{w}$ vector to $U_1 + 5sgn(U_2)$, where U_1 and U_2 were generated from a standard Normal distribution. For each node j, we included each other node in its parent set π_j with probability 0.25. To generate samples that approximate the equilibrium distribution of the model, we started by sampling each node's value based on its bias w_0 alone, then we performed 1000 updates, where in each update we updated all nodes whose parents were updated in the previous iteration. We assume perfect interventions, which force a variable into a given state. In the special case where the dependency structure between the nodes is acyclic, this sampling procedure is exactly equivalent to ancestral sampling in a DAG model (and the update distributions are the corresponding conditional distributions), and these interventions are equivalent to perfect interventions in the DAG. However, we do not enforce acyclicity, so the distribution may have feedback cycles (which are common in biological networks).

We considered 2 variants of this data, one where all variables are visible, and one with hidden variables (as is common in most real problems). In the *visible* SEM data set, we generated from an 8-node SEM model under all 28 pairs of action combinations. In our experiments, we trained on 27 of the action pairs and tested on the remaining action pair, for all 28 pairs. In the *hidden* SEM data set, we generated from a 16-node SEM model under the 28 pairs of actions combinations for the first 8 nodes, but we treat the odd-numbered half of the nodes as hidden (so half of the actions affect a visible node in the model, and half of the actions affect a hidden node). We think that this is a slightly more realistic synthetic data set than a fully visible DAG with perfect interventions, due to the presence of hidden nodes and feedback cycles, as well as interventions that affect both visible and hidden nodes. When the data is visualized, it looks qualitatively similar to the T-cell data in Figure 1 (results not shown).

The results on the visible data are shown in Figure 8. Since we are only testing on new action combinations, independent models cannot be applied. As expected, conditional models do better than ignore models. However, amongst the conditional models there does not appear to be a clear winner. In particular, DAG models, even perfect DAGs which are told the target of intervention, do no better than non-DAG models.

The results on the hidden data are not shown, since they are qualitatively similar to the visible case. Note that in this setting, we cannot use the perfect intervention model, since some of the interventions affected hidden nodes; hence the target of intervention is not well defined. We have obtained qualitatively similar results on other kinds of synthetic data.

4 Conclusions

In this paper, we have argued that it is helpful to think of causal models in functional terms, and to evaluate them in terms of their predictive performance, rather than in terms of graph structures that they learn. In particular, we view causal modeling as equivalent to learning a conditional density model of the form $p(\mathbf{x}|\mathbf{a})$.

A criticism of this work could be that we are not really doing causality because we can't predict the effects of new actions. However, in general, this is impossible unless we know something (or assume something) about the new action, since in general $p(\mathbf{x}|a_1 = 1, a_2 = 0)$ need have nothing to do with $p(\mathbf{x}|a_1 = 0, a_2 = 1)$. Indeed, when we tested the ability of various methods, including causal DAGs, to predict the effects of a novel action in the T-cell data, they all performed poorly — not significantly better than methods which ignore the actions altogether. This is despite the fact that the DAG structure we were using was the MAP optimal DAG, which had previously been shown to be close to the "true" structure, and that we knew what the targets of the novel action were.

We think a promising direction for future work is to describe actions, and/or the variables they act on, in terms of feature vectors, rather than treating them as atomic symbols. This transforms the task of predicting the effects of new actions into a standard structured prediction problem, that could be addressed with CRFs, M3Ns, etc. Just like predicting the labeling of a new sentence or image given only its features, if there is some regularity in the action-feature space, then we can predict the effects of a new action given only the features of the action, without ever having to perform it.

Acknowledgments

We would like to thank Guillaume Alain for help with some of the experiments, and Gustavo Lacerda and Oliver Schulte for comments on a draft version of this paper.

References

O. Banerjee, L. El Ghaoui, and A. d'Aspremont. Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. *J. of Machine Learning Research*, 9: 485–516, 2008.

G. Cooper and C. Yoo. Causal discovery from a mixture of experimental and observational data. In *UAI*, 1999.

A. P. Dawid. Influence diagrams for causal modelling and inference. *Intl. Stat. Review*, 70:161–189, 2002. Corrections p437.

A. P. Dawid. Beware of the DAG! *J. of Machine Learning Research*, 2009. To appear.

J. Duchi, S. Gould, and D. Koller. Projected subgradient methods for learning sparse gaussians. In *UAI*, 2008.

D. Eaton and K. Murphy. Exact Bayesian structure learning from uncertain interventions. In *AI/Statistics*, 2007.

F. Eberhardt, C. Glymour, and R. Scheines. On the number of experiments sufficient and in the worst case necessary to identify all causal relations among N variables. In *UAI*, 2005.

F. Eberhardt, C. Glymour, and R. Scheines. N-1 experiments suffice to determine the causal relations among N variables. In *Innovations in Machine Learning*. Springer, 2006.

J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation the graphical lasso. *Biostatistics*, 2007.

M. I. Jordan and R. A. Jacobs. Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, 6:181–214, 1994.

S. Lauritzen. Causal inference from graphical models. In D. R. Cox O. E. Barndoff-Nielsen and C. Klueppelberg, editors, *Complex stochastic systems*. 2000.

S. Lauritzen and T. Richardson. Chain graph models and their causal interpretations. *J. of the Am. Stat. Assoc.*, 3(64):321–361, 2002.

S.-I. Lee, V. Ganapathi, and D. Koller. Efficient structure learning of Markov networks using L1-regularization. In *NIPS*, 2006.

N. Meinshausen and P. Buhlmann. High dimensional graphs and variable selection with the lasso. *The Annals of Statistics*, 34:1436–1462, 2006.

J. Pearl. *Causality: Models, Reasoning and Inference*. Cambridge Univ. Press, 2000.

K. Sachs, O. Perez, D. Pe'er, D. Lauffenburger, and G. Nolan. Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308, 2005.

R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems*, volume 20, 2008.

M. Schmidt, K. Murphy, G. Fung, and R. Rosales. Structure Learning in Random Fields for Heart Motion Abnormality Detection. In *CVPR*, 2008.

T. Silander and P. Myllmaki. A simple approach for finding the globally optimal Bayesian network structure. In *UAI*, 2006.

P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search*. MIT Press, 2000. 2nd edition.

M. Wainwright, P. Ravikumar, and J. Lafferty. Inferring graphical model structure using ℓ_1-regularized pseudo-likelihood. In *NIPS*, 2006.