# Exact Bayesian structure learning from uncertain interventions

**Daniel Eaton**
Computer Science Dept.
University of British Columbia
deaton@cs.ubc.ca

**Kevin Murphy**
Computer Science Dept.
University of British Columbia
murphyk@cs.ubc.ca

## Abstract

We show how to apply the dynamic programming algorithm of Koivisto and Sood [KS04, Koi06], which computes the exact posterior marginal edge probabilities $p(G_{ij} = 1|D)$ of a DAG $G$ given data $D$, to the case where the data is obtained by interventions (experiments). In particular, we consider the case where the targets of the interventions are a priori unknown. We show that it is possible to learn the targets of intervention at the same time as learning the causal structure. We apply our exact technique to a biological data set that had previously been analyzed using MCMC [SPP$^+$05, EW06, WGH06].

## 1 Introduction

The use of Bayesian networks to represent causal models has become increasingly popular [Pea00, SGS00]. In particular, there is much interest in learning the structure of these models from data. Given observational data, it is only possible to identify the structure up to Markov equivalence. For example, the three models $X \rightarrow Y \rightarrow Z$, $X \leftarrow Y \leftarrow Z$, and $X \leftarrow Y \rightarrow Z$ all encode the same conditional independency statement, $X \perp Z|Y$. To distinguish between such models, we need interventional (experimental) data [EGS05].

Most previous work has focused on the case of "perfect" interventions, in which it is assumed that an intervention sets a single variable to a specific state (as in a randomized experiment). This is the basis of Pearl's "do-calculus" (as in the verb "to do") [Pea00]. A perfect intervention essentially "cuts off" the influence of the parents to the intervened node, and can be modeled as a structural change by performing "graph surgery" (removing incoming edges from the intervened node). Although some real-world interventions can be modeled in this way (such as gene knockouts), most interventions are not so precise in their effects.

One possible relaxation of this model is to assume that interventions are "stochastic", meaning that they induce a distribution over states rather than a specific state [KHNA04]. A further relaxation is to assume that the effect of an intervention does not render the node independent of its parents, but simply changes the parameters of the local distribution; this has been called a "mechanism change" [TP01b, TP01a] or "parametric change" [EGS06]. For many situations, this is a more realistic model than perfect interventions, since it is often impossible to force variables into specific states.

In this paper, we propose a further relaxation of the notion of intervention, and consider the case where the targets of intervention are uncertain. This extension is motivated by problems in molecular biology, where the effects of various chemicals that are added are not precisely known. In particular, each chemical may affect a hidden variable, which can in turn affect multiple observed variables, often in unknown ways. We model this by adding the intervention nodes to the graph, and then performing structure learning in this extended, two-layered graph.

Our contributions are three-fold. First, we show how to combine models of intervention — perfect, imperfect and uncertain — with a recently proposed algorithm for efficiently determining the exact posterior probabilities of the edges in a graph [KS04, Koi06]. Second, we show empirically that it is possible to infer the true causal graph structure, even when the targets of interventions are uncertain, provided the interventions are able to affect enough nodes. Third, we apply our exact methodology to a biological dataset that had previously been analyzed using MCMC [SPP$^+$05, EW06].

## 2 Models of intervention

We will first describe our probability model under the assumption that there are no interventions. Then we will describe ways to model the many kinds of interventions that have been proposed in the literature, culminating in our model of uncertain interventions. This will serve to situate our model in the context of previous work.

### 2.1 No interventions

For the intervention-free case, we will assume that the conditional probability distribution (CPD) of each node in the graph is given by $p(X_i|X_{G_i}, \theta, G) = f_i(X_i|X_{G_i}, \theta_i)$, where $G_i$ are the parents of $i$ in $G$, $\theta_i$ are $i$'s parameters, and $f_i()$ is some probability density function (e.g., multinomial or linear Gaussian). For the parameter prior $p(\theta|G)$, we will make the usual assumptions of global and local independence, and parameter modularity (see [HGC95] for details). We will further assume that each $p(\theta_i)$ is conjugate to $f_i$, which allows for closed form computation of the marginal likelihood $p(X^{1:N}|G) = \int p(X^{1:N}|G, \theta)p(\theta)d\theta$, where $N$ is the number of data cases. For example, for multinomial-Dirichlet, the marginal likelihood for a family (a node and its parents) is given by [HGC95]

$$\begin{aligned} p(x_i^{1:N}|x_{G_i}^{1:N}) &= \int [\prod_{n=1}^{N} p(x_i^n|x_{G_i}^n, \theta_i)]p(\theta_i)d\theta_i \\ &= \prod_{j=1}^{r_i} \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + N_{ij})} \prod_{k=1}^{q_i} \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})} \end{aligned}$$

where $N_{ijk} = \sum_{n=1}^{N} I(x_i^n = k, x_{G_i}^n = j)$ are the counts, and $N_{ij} = \sum_k N_{ijk}$. ($I(e)$ is the indicator function in which $I(e) = 1$ if event $e$ is true and $I(e) = 0$ otherwise.) Also, $\alpha_{ijk}$ are the pseudo counts (Dirichlet hyper parameters), $\alpha_{ij} = \sum_k \alpha_{ijk}$, $r_i$ is the number of discrete states for $X_i$, and $q_i$ is the number of states for $X_{G_i}$. We will usually use the BDeu prior $\alpha_{ijk} = 1/q_i r_i$ [HGC95]. (An analogous formula can be derived for the normal-Gamma case [GH02].) The marginal likelihood of all the nodes is then given by $p(X^{1:N}|G) = \prod_{i=1}^{d} p(X_i^{1:N}|X_{G_i}^{1:N})$, where $d$ is the number of nodes.

### 2.2 Perfect interventions

If we perform a perfect intervention on node $i$ in data case $n$, then we set $X_i^n = x_i^*$, where $x_i^*$ is the desired "target state" for node $i$ (assumed to be fixed and known). We modify the CPD for this case to be $p(X_i|X_{G_i}, \theta) = I(X_i = x_i^*)$. We see that $X_i$ is effectively "cut off" from its parents $X_{G_i}$.

### 2.3 Imperfect interventions

A simple way to model interventions is to introduce intervention nodes, that act like "switching parents": if $I_i^n = 1$, then we have performed an intervention on node $i$ in case $n$ and we use a different set of parameters than if $I_i^n = 0$, when we use the "normal" parameters. Specifically, we set $p(X_i|X_{G_i}, I_i = 0, \theta, G) = f_i(X_i|X_{G_i}, \theta_i^0)$ and $p(X_i|X_{G_i}, I_i = 1, \theta, G) = f_i(X_i|X_{G_i}, \theta_i^1)$. (Note that the assumption that the functional form $f_i$ does not change is made without loss of generality, since $\theta_i$ can encode within it the specific type of function.) Tian and Pearl [TP01b, TP01a] refer to this as a "mechanism change": see Figure 1. A special case of this is a perfect intervention, in which $p(X_i|X_{G_i}, I_i = 1, \theta, g) = I(X_i = x_i^*)$. To simplify notation, we assume every node has its own intervention node; if a node $i$ is not intervenable, we simply clamp $I_i^n = 0$ for all $n$.

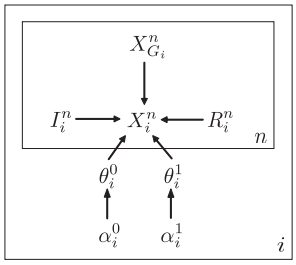


*Figure 1:* Model of mechanism change. $X_i^n$ is node $i$ in case $n$, $X_{G_i}^n$ are its parents. $I_i^n$ acts like a switching variable: If $I_i^n = 1$ (representing an intervention), then $X_i$ uses the parameters $\theta_i^1$; If $I_i^n = 0$, then $X_i$ uses the parameters $\theta_i^0$. $\alpha_i^{0/1}$ are the hyper-parameters. We can optionally add another switch node $R_i^n$, which can be used to model the degree of effectiveness of the intervention (see text for details).

When we have interventional data, we modify the local marginal likelihood formula by partitioning the data into those cases in which $X_i$ was passively observed, and those in which $X_i$ was set by intervention:

$$\begin{aligned} p(x_i^{1:N}|x_{G_i}^{1:N}, I_i^{1:N}) &= \int [\prod_{n:I_i^n=0} p(x_i^n|x_{G_i}, \theta_i^0)]p(\theta_i^0)d\theta_i^0 \\ &\times \int [\prod_{n:I_i^n=1} p(x_i^n|x_{G_i}, \theta_i^1)]p(\theta_i^1)d\theta_i^1 \end{aligned}$$

In the case of perfect interventions, this second factor evaluates to 1, so we can simply drop cases in which node $i$ was set by intervention from the computation of the marginal likelihood of that node [CY99].

We can also model the case where the interventions are unreliable, by introducing a latent indicator $R_i^n$, where $R_i^n = 1$ means the intervention succeeded, and $R_i^n = 0$ means it failed. In this case, $p(X_i|X_{G_i}, \theta, I_i = 1)$ becomes a mixture model. The prior mixture weight $p(R_i = 1)$ is the "effectiveness" of the intervention [KHNA04].

Another way to model imperfect interventions is as "soft" interventions, in which an intervention just increases the likelihood that a node enters its target state $x_i^*$. Markowetz et al. [MGR05] suggest using the same model of $p(X_i|X_{G_i}, I_i, \theta, G)$ as before, but now the parameters $\theta_i^0$ and $\theta_i^1$ have *dependent* hyper-parameters. In particular, for the multinomial-Dirichlet case, $\theta_{ij\cdot}^{0/1} \sim Dir(\alpha_{ij\cdot}^{0/1})$, they assume the deterministic relation $\alpha_{ij\cdot}^1 = \alpha_{ij\cdot}^0 + w_i \vec{e}_t$, where $j$ indexes states (conditioning cases) of $x_{G_i}$, $t = x_i^*$ is the target value for node $i$, $\vec{e}_t = (0, \ldots, 0, 1, 0, \ldots, 0)$ with a 1 in the $t$'th position, and $w_i$ is the strength of the interven-

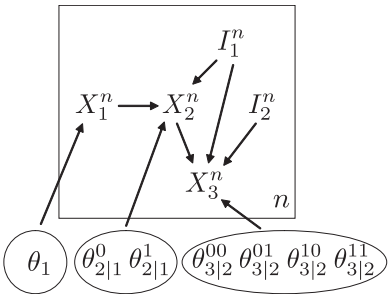


*Figure 2:* An example of "fat hand" interventions. Intervention 1 affects nodes 2 and 3, intervention 2 affects node 3. The parameters for node 3 are $\theta_{3|2}^{ij}(k,\ell)$, where $I_1 = i$, $I_2 = j$, $X_2 = k$ and $X_3 = \ell$.

tion. As $w_i \to \infty$, this becomes a perfect intervention.

### 2.4 Uncertain interventions

Finally we come to our proposed model for representing interventions with uncertain targets, as well as uncertain effects. We no longer assume a one to one correspondence between intervention nodes $I_i$ and "regular" nodes $X_i$. Instead, we assume that each intervention node $I_i$ may have multiple regular children. (Such interventions are sometimes said to be due to a "fat hand", which "touches" many variables at once.) If a regular node has multiple intervention parents, we create a new parameter vector for each possible combination of intervention parents: see Figure 2 for an example.

We are interested in learning the connections from the intervention nodes to the regular nodes, as well as between the regular nodes. We do not allow connections between the intervention nodes, or from the regular nodes back to the intervention nodes, since we assume the intervention nodes are exogeneous and fixed.

To explain how we modify the marginal likelihood function, we need some more notation. Let $X_{G_i}$ be the regular parents of node $i$, and $I_{G_i}$ be the intervention parents. Let $\theta_i^\ell$ be the parameters for node $i$ given that its intervention parents have state $\ell$. Then the marginal likelihood for a family becomes

$$p(x_i^{1:N} | x_{G_i}^{1:N}, I_{G_i}^{1:N}) = \prod_\ell \int \left[ \prod_{n: I_{G_i}^n = \ell} p(x_i^n | x_{G_i}^n, \theta_i^\ell) \right] p(\theta_i^\ell) d\theta_i^\ell$$

### 2.5 The power of interventions

The ability to recover the true causal structure (assuming no latent variables) using perfect and imperfect interventions has already been demonstrated both theoretically [EGS05, EGS06, TP01a, TP01b] and empirically

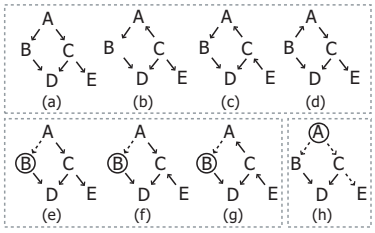


*Figure 3:* Top left: the "cancer network", from [FMR98]. (a-d) are Markov equivalent. (c-g) are equivalent under an intervention on $B$. (h) is the unique member under an intervention on $A$. Based on [TP01b].

[CY99, MS03, TP01a, TP01b, WGH06]. Specifically, each intervention determines the direction of the edges between the intervened nodes and its neighbors; this in turn may result in the direction of other edges being "compelled" [Chi95].

For example, in Figure 3, we see that there are 4 graphs that are Markov equivalent to the true structure; given observational data alone, this is all we can infer. However, given enough interventions (perfect or imperfect) on $B$, we can eliminate the fourth graph (d), since it has the wrong parents for $B$. Given enough interventions on $A$, we can uniquely identify the graph, since we can identify the arcs out of A by intervention, the arcs into D since it is a v-structure, and the $C \to E$ arc since it is compelled. In general, given a set of interventions and observational data, we can identify a graph up to intervention equivalence (see [TP01a] for a precise definition).

In Section 4.1, we will experimentally study the question of whether one can still learn the true structure from uncertain interventions (i.e., when the targets of intervention are a priori unknown), and if so, how much more data one needs compared to the case where the intervention targets are known.

## 3 Algorithms for structure learning

The Bayesian approach to structure learning avoids many of the conceptual problems that arise when trying to combine the results of potentially inconsistent conditional independency tests performed on different ("mutated") models [Ebe06]. In addition, it is particularly appropriate when the sample sizes are small, but "soft" prior knowledge is available, as in many molecular biology experiments.

However, we are left with a computational problem. Computing the full posterior is intractable, since there are $O(d! 2^{\binom{d}{2}})$ DAGs (directed acyclic graphs) on $d$ nodes [Rob73].[1] So all one can realistically hope to do is to

[1] The exact formula is given by the following recurrence equation: $r(d) = \sum_{i=1}^{d} (-1)^{i+1} \binom{d}{i} 2^{i(d-i)} r(d-i)$. This gives $r(2) = 3$, $r(3) = 25$, $r(4) = 543$, $r(5) = 29,281$, $r(6) =$

compute the posterior probability of certain features of the graph using Bayesian model averaging:

$$p(f|D) = \sum_G p(G|D) f(G)$$

where $f(G) = 1$ if graph $G$ has the feature (e.g., an edge from $i$ to $j$), and $f(G) = 0$ otherwise. (In the small sample regime, the posterior over models often has many modes, so it would be unwise to pick any single model, assuming one's goal is scientific discovery.)

Standard MCMC methods for sampling from the posterior (see e.g., [MY95]) are very slow and do not mix well, due to the size of the search space and the "peakiness" of the posterior landscape. A significant advance was made by Friedman and Koller [FK03], who suggested sampling over the space of node orderings, which "only" has size $O(d!)$. Koivisto and Sood [KS04, Koi06] made another significant advance, by showing that one can compute the exact posterior probabilities of all edges using dynamic programming (DP) in $O(d2^d)$ time, essentially by summing over all node orderings instead of sampling them. While still exponential in $d$, this is significantly better than $O(d!2^{d^2})$, and allows exact analysis of models with up to about $d = 20$ variables.

The DP algorithm is rather complex, and we do not have space to explain it here. For the purposes of this paper, it suffices to know that the input to the algorithm is a prior over node orderings $q_i(U_i)$, a prior over possible parent sets, $\rho_i(G_i)$, and a local marginal likelihood function for every node and every possible parent set, $p(X_i|X_{G_i})$. We discuss each of these in turn below. We then discuss extensions to the algorithm to handle interventions.

### 3.1 Priors

A node ordering $\prec$ may be specified by the vector $(U_1, \ldots, U_d)$, where $U_i = \{j : j \prec i\}$ are the set of nodes that preceed $i$. Following [KS04], we will assume a uniform prior over orderings, $q_i(U_i) \propto 1$.

A parent set may be specified by the vector $G_i \subset V$, where $V$ is the set of nodes. Note that this is an unordered set; the ordering of the elements is specified by $U_i$. Following [KS04], we set $\rho_i(G_i) \propto \binom{d-1}{|G_i|}^{-1}$, if $|G_i| \leq k$, and $\rho_i(G_i) = 0$ otherwise, where $k$ is a fan-in bound for each node. (By setting $k = d - 1$, we can eliminate the fan-in restriction.)

Of course, $G_i$ and $U_i$ are not independent, since we require $G_i \subseteq U_i$. Hence $q_i(U_i)$ and $\rho_i(G_i)$ should not be thought of as probabilities, but rather as potential functions or factors, which jointly define the prior over orderings and graphs as follows

$$p(\prec, G) \quad = \quad \frac{1}{Z} \prod_{i=1}^{d} q_i(U_i) \rho_i(G_i) \times I(\prec, G \text{ consistent})$$

where the last term checks that $G$ is consistent with $\prec$, and that $\prec$ is a total order (and hence $G$ is acyclic). $Z$ is a normalization constant which will cancel out when computing posterior features. By marginalizing over $\prec$, we induce a prior over graphs $p(G)$. The induced prior is highly non uniform, but favors sparse graphs, since parent sets that are smaller are consistent with more orderings and therefore more probable.

The reason the prior is defined in this indirect way is that the dynamic programming algorithm relies on the fact that we can compute the score for certain parent sets without knowing what the order of those parents are; hence we can re-use that score for all orderings of the parents. See [KS04, FK03, EW06] for a more detailed discussion of the relationship between priors on orders and graphs.

### 3.2 Likelihoods

The final inputs to the algorithm are the local conditional marginal likelihoods $p(x_i^{1:N}|x_{G_i}^{1:N}, I_{G_i}^{1:N})$, which must be computed for every node $i$ and every possible parent set $G_i$ (up to size $k$). There are $\binom{d}{k} = O(d^k)$ such terms. The cost of computing each term depends on the form of the local CPDs $f_i$ and the prior $p(\theta_i)$. We have already given the formula for the multinomial-Dirichlet case. It takes $O(N)$ time to compute the sufficient statistics (counts) $N_{ijk}$, where $N$ is the number of training cases. We have found that 95% of the overall algorithm time is spent computing these terms, even for relatively small ($N \sim 5000$) datasets. Fortunately, one can use AD-trees [ML98] to speed this up.

### 3.3 Layering

In the case where we include the intervention nodes in the graph, we use a two layered graph structure, $V = \mathcal{X} \cup \mathcal{I}$, where $\mathcal{X}$ are the regular nodes and $\mathcal{I}$ are the intervention nodes. The prior ensures there are no edges between the $\mathcal{I}$ nodes, and no edges from $\mathcal{X}$ back to $\mathcal{I}$. Let $d_I = |\mathcal{I}|$ be the number of intervention nodes, and $d_X = |\mathcal{X}|$ be the number of regular nodes. The time complexity of the DP algorithm in this case is $O(d2^{d_X} + d^{k+1}C(N))$, where $d = d_I + d_X$, and $C(N)$ is the cost of computing each local marginal likelihood term. Note that layering is crucial for efficiently handling uncertain interventions, otherwise the algorithm would take $O(d2^d)$ instead of $O(d2^{d_X})$ time.

## 4 Experimental results

We first present some results on synthetic data generated from a Bayes net of known structure, and then present re-

$3,781,503$, $r(7) = 1.1 \times 10^9$, etc.

sults on a real biological data set.

### 4.1 Synthetic data

In this section, we experimentally study the question of whether one can still learn the true structure, even when the targets of intervention are a priori unknown, and if so, how much more data one needs compared to the case where the intervention targets are known.[2] We assessed this using the following experimental protocol. We considered the graph structure in Figure 3, and then generated random multinomial CPDs by sampling from a Dirichlet distribution with hyper-parameters chosen by the method described in [CM02]. This ensures that there are reasonably strong dependencies between the nodes. (We used binary nodes for simplicity.) We then generated data using forwards sampling; the first 2000 cases $D_0$ were from the original model, the second 2000 cases $D_1$ from a "mutated" model, in which we performed a perfect intervention either on $A$ or $B$, forcing it to the "off" state in each case.

Next we tried to learn back the structure using varying sample sizes of $N \in \{100, 500, 2000\}$. Specifically we used $N$ observational samples and $N$ interventional samples, $D = (D_0^{1:N}, D_1^{1:N})$. We ran the algorithm using data $D$ and under increasingly vague prior knowledge: (1) using the perfect interventions model; (2) using the soft interventions model[3]; (3) using the imperfect (mechanism change) model; and (4) using the uncertain interventions model. In the latter case, we also learned the children of the intervention node. As a control, we also tried just using observational data, $D = D_0^{1:2N}$.

Our results are shown in Figure 4. We see that with observational data alone, we are only able to recover the v-structure $B \rightarrow D \leftarrow C$, with the directions of the other arcs being uncertain (e.g., $P(C \rightarrow E) \approx 0.75$.) With perfect interventions on $B$, we can additionally recover the $A \rightarrow B$ arc, and with perfect interventions on $A$, we can recover the graph uniquely, consistent with the theoretical results in Section 2.5. With imperfect and soft interventions, we need somewhat more data, but results are otherwise very similar to the perfect case, and are omitted due to lack of space. With uncertain interventions, we see that the entropy of the posterior on the regular edges is higher than when using perfect interventions, but it too reduces with sample size. Eventually the posterior converges to a delta function on the intervention equivalence class. We obtain similar results with other experiments on random graphs.

[2] Tian and Pearl [TP01a] briefly mention the case of "unknown focal variables" (which we are calling uncertain targets of intervention) in the context of constraint based learning methods, but do not present any algorithms for identifying focal variables. We are not aware of any other papers that address this question.

[3] [MGR05] do not discuss how to set the pushing strength $w_i$. We set it equal to $0.5N$, so that the data does not overwhelm the hyper-parameter $\alpha_{ijk}^1$.

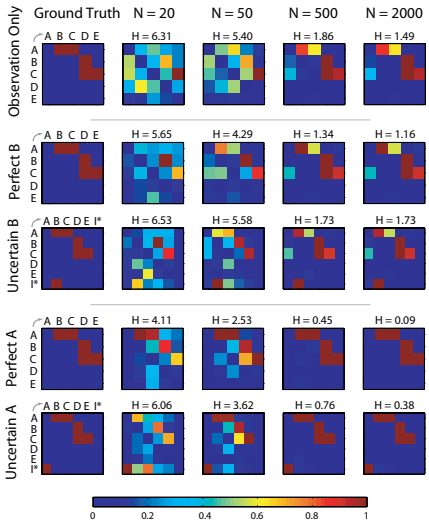


*Figure 4:* Results of structure learning on the cancer network (Figure 3). Left column: ground truth. Subsequent columns: posterior edge probabilities $p(G_{ij} = 1|D)$ for increasing sample sizes $N$, where dark red denotes 1.0 and dark blue denotes 0.0. $H$ is the entropy of the factored posterior $\prod_{ij} p(G_{ij}|D)$. See text for details. This figure is best viewed in colour.

This suggests that our proposed mechanism is easily able to learn causal structure even from uncertain interventions.

### 4.2 Biological data

We now apply our methodology to a real biological data set, which had previously been analyzed using MCMC by Sachs et al [SPP+05] (who used multiple restart simulated annealing in the space of DAGs), Werhli et al. [WGH06] (who used Metropolis Hastings in the space of node orderings), and Ellis and Wong [EW06] (who used equi-energy sampling in the space of node orderings). The purpose of our experiment is to determine the exact posterior over edges, and hence to assess the quality of the MCMC techniques, and also to learn the effects of the interventions that were performed.

The dataset consists of 11 protein concentration levels measured under 6 different interventions, plus 3 unperturbed measurements. The proteins in question constitute part of the signaling network of human T-cells, and therefore play a vital role in the immune system. See Figure 6(a) for a depiction of the commonly accepted "ground truth" network, including hidden nodes.

The data in question were gathered using a technique called flow cytometry, which can record phosphorylation levels of individual cells. This has two advantages compared to other measurement techniques: first, it avoids the information loss commonly incurred by averaging over ensembles of cells; second, it creates relatively large sample sizes (we have $N = 5400$ data points in total, 600 per condition).

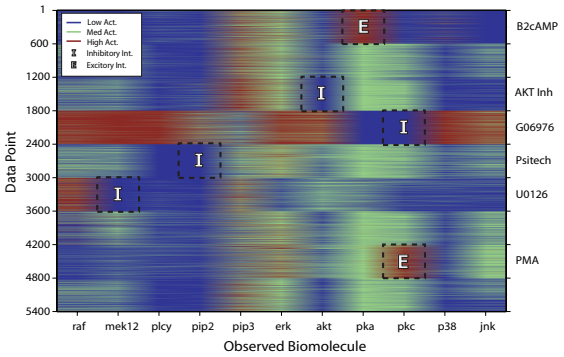


*Figure 5:* Discretized biological data from [SPP$^+$05]. Columns are the 11 measured proteins, rows are the 9 experimental conditions, 3 of which are "general stimulation" rather than specific interventions. The name of the chemical that was added in each case is shown on the right. The intended primary target is indicated by an E (for excitation) or I (for inhibition). This figure is best viewed in colour.

The raw data was discretized into 3 states, representing low, medium and high activity. We obtained this discretized data directly from Sachs; see Figure 5 for a visualization. This constituted the input to our algorithm.

We tried two different analyses. In the first version, we assumed that the targets of intervention were known, and we modeled these using perfect interventions (as did Sachs et al). The results are shown in Figure 6(c). These should be compared with the results of the MCMC analysis of Sachs et al, which are shown in Figure 6(b), and the ground truth network, which is shown in Figure 6(a).

While there is substantial agreement between the three models, there are also many differences. For example, the ground truth shows no edge from jnk to p38, or from mek12 to jnk, yet both inference methods detect such an edge. This may be due to the presence of various hidden variables. Looking at the data in Figure 5, mek12 and jnk seem quite highly correlated, although this is obviously not enough evidence to suggest there should be an edge between them (as shown in [SPP$^+$05], nearly all of the variables are significantly pairwise correlated!).

There are also several edges in our model that seem to be absent in the MCMC analysis of Sachs et al. (denoted by dashed edges). This is possibly because Sachs et al only perform model averaging over a "compendia of high scoring networks", as found by 500 restarts of simulated annealing, whereas our method averages over all graphs, and hence may detect support for many more edges. (Note that averaging over many sparse, but different, graphs can result in a dense set of marginal edge probabilities.) Also, the two methods use different graph priors $p(G)$, and hence cannot be directly compared.

In the second experiment, we added the intervention nodes to the graph and learned their children, rather than prespecifying them. The results are shown in Figure 6(d). We successfuly identified the known targets of all but one of the 6 interventions. (We missed the G06967 $\rightarrow$ pkc edge.) However, we also found that the interventions have multiple children, even though they were designed to target specific proteins. Upon further investigation, we found that each intervention typically affected a node and some of its immediate neighbors. For example, from the ground truth network in Figure 6(a), we see that Psitect (designated 8 in that figure) is known to inhibit pip2; in our learned network (Figure 6(d)), we see that Psitect connects to pip2, but also to plcy, which is a neighbor of pip2. This is biologically plausible, since some of these interventions actually work by altering hidden variables, which can therefore cause changes in several neighboring visible variables. Also, although we missed the G06967 $\rightarrow$ pkc edge, the other children of G06967 (plcy, pka, mek12, erk and p38) seem to be strongly affected by G06967 when looking at the data in Figure 5.

We also tried analysing the continuous data using linear-Gaussian Bayes nets [GH02]. Following [EW06], we took a log transform of each variable and then standardized them. Our results (omitted due to lack of space) are similar to [EW06], but our graph is much denser, suggesting that their MCMC scheme failed to visit sufficiently many modes. (Although once again our results are not directly comparable due to the different prior.) The graphs inferred using the Gaussian and multinomial models have much in common, but they also differ in many of the details. A discussion of which model is more appropriate is beyond the scope of this paper.

It is difficult to rigorously assess the quality of our results when there is no ground truth. (The biological model in Figure 6(a) is unlikely to be the "true" model that generated the data in Figure 5. Also, it contains hidden variables, so is not directly comparable to what we are learning.) The approach taken by Ellis et al [EW06] was to compare the predictive log-likelihood in a cross-validation framework. This can also be done using the DP algorithm, by computing $p(x|D) = p(x, D)/p(D)$; these normalization constants can be obtained by running the "forwards" algorithm of [KS04] using the "dummy" feature $f = 1$. We are currently performing this experiment. However, this is quite slow, since we need to rerun the algorithm for every test point $x$.

### 4.3 Running time

For the 3-state biological data, with $d = 11$ nodes (using perfect interventions) and $N = 5400$, our Matlab implementation only took 30 seconds.[4] For the case where we learned the effects of interventions (so $d = 17$), it took

[4] Experiments were performed on a laptop with a 2 GHz Intel Core Duo Processor and 2GB RAM running under Windows XP.

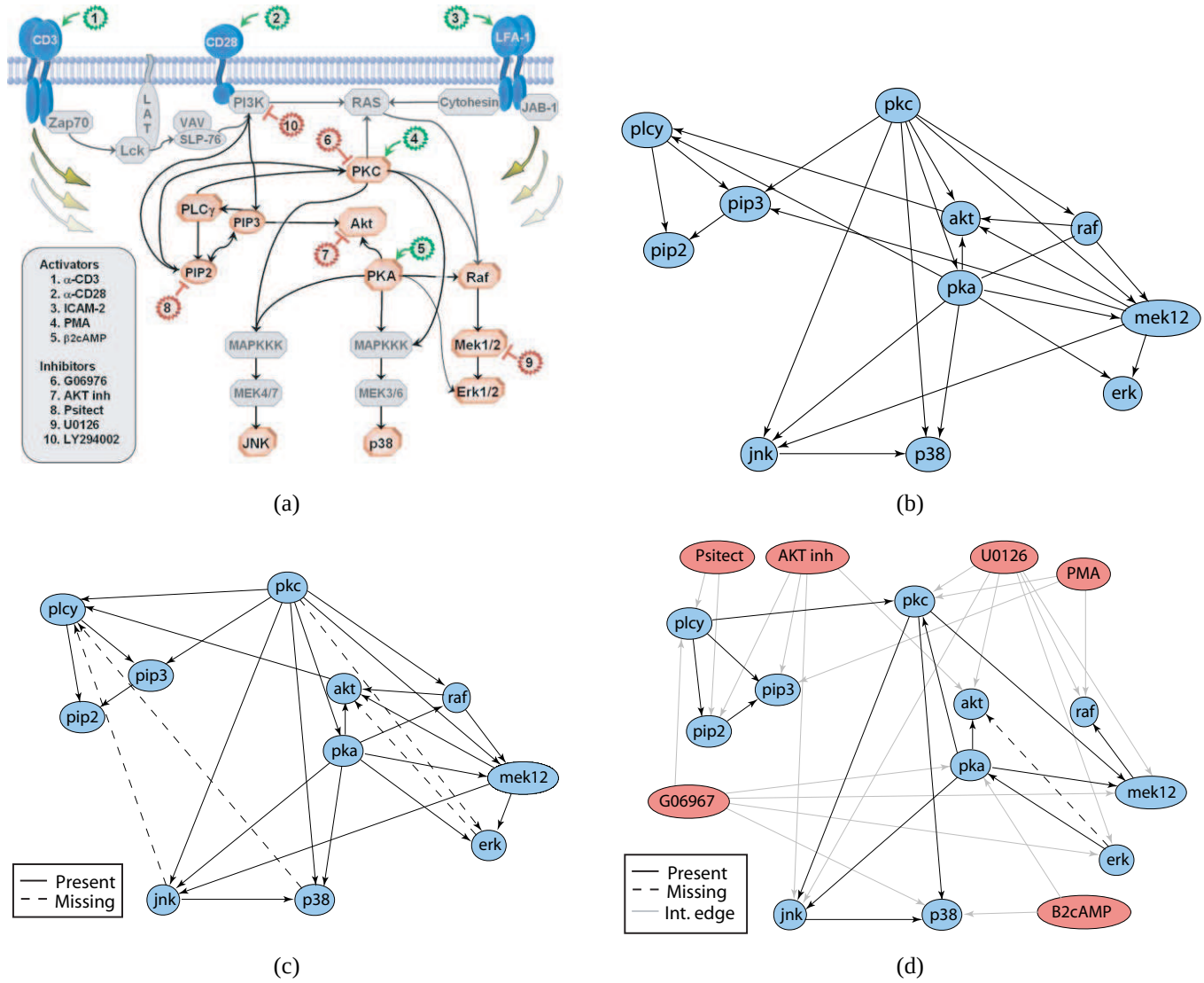


*Figure 6:* Models of the biological data. (a) A partial model of the T-cell pathway, as currently accepted by biologists. The small round circles with numbers represent various interventions (green = activators, red = inhibitors). From [SPP$^+$05]. Reprinted with permission from AAAS. (b) Edges with marginal probability above 0.5 as estimated by [SPP$^+$05]. (c) Edges with marginal probability above 0.5 as estimated by us, assuming known perfect interventions. Dashed edges are ones that are missing from the union of (a) and (b). These are either false positives, or edges that Sachs et al missed. (d) Edges with marginal probability above 0.5 as estimated by us, assuming uncertain, imperfect interventions, and a fan-in bound of $k = 2$. The intervention nodes are in red, and edges from the intervention nodes are light gray. Dashed edges are ones that are missing from the union of (a) and (b). This figure is best viewed in colour.

about 30 minutes (using a fan-in bound of $k = 2$).

## 5 Summary and future work

We have shown how to apply the dynamic programming algorithm of Koivisto and Sood [KS04, Koi06] to learn causal structure from interventional data. The main bottleneck to tackling larger problems is the space and time limit of $O(d2^d)$, which limits us to about $d = 20$. However, one can exploit the layering idea to extend this to much larger graphs. (See [MKTG06] for some ideas on how to partition nodes into groups/ layers in an unsupervised way.) Layering should also enable the learning of dynamic Bayes nets (DBNs) [FMR98].

Another issue that deserves more attention is the non-uniform prior $p(G)$ that the DP algorithm implictly uses. It would be useful if one could use an arbitrary prior on graphs. We are currently developing a method where we use the output of the DP algorithm as a proposal distribution for a Metropolis Hastings search through DAG space; this lets us use an arbitrary graph prior, but works much better than standard proposal distributions. Finally, it would be interesting to extend the ideas in this paper to the active learning case [Mur01, TK01], where one has to decide which interventions to perform.

**Acknowledgements**

We would like to thank Karen Sachs and Byron Ellis for sharing their data with us, and Mikko Koivisto for sending us a pre-release version of his C++ code.

## References

[Chi95] D. Chickering. A transformational characterization of equivalent Bayesian network structures. In *UAI*, 1995.

[CM02] D. Chickering and C. Meek. Finding Optimal Bayesian Networks. In *UAI*, 2002.

[CY99] G. Cooper and C. Yoo. Causal discovery from a mixture of experimental and observational data. In *UAI*, 1999.

[Ebe06] F. Eberhardt. Sufficient condition for pooling data from different distributions. In *First Symposium on Philosophy, History, and Methodology of Error*, 2006.

[EGS05] 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.

[EGS06] F. Eberhardt, C. Glymour, and R. Scheines. Interventions and causal inference. In *20th Mtg. Philos. of Sci. Assoc.*, 2006.

[EW06] B. Ellis and W. Wong. Sampling Bayesian Networks quickly. In *Interface*, 2006.

[FK03] N. Friedman and D. Koller. Being Bayesian about Network Structure: A Bayesian Approach to Structure Discovery in Bayesian Networks. *Machine Learning*, 50:95–126, 2003.

[FMR98] N. Friedman, K. Murphy, and S. Russell. Learning the structure of dynamic probabilistic networks. In *UAI*, 1998.

[GH02] D. Geiger and D. Heckerman. Parameter priors for directed acyclic graphical models and the characterization of several probability distributions. *The Annals of Statistics*, 30(5):1412–1440, 2002.

[HGC95] D. Heckerman, D. Geiger, and M. Chickering. Learning Bayesian networks: the combination of knowledge and statistical data. *Machine Learning*, 20(3):197–243, 1995.

[KHNA04] K. Korb, L. Hope, A. Nicholson, and K. Axnick. Varieties of causal intervention. In *Pacific Rim Conference on AI*, 2004.

[Koi06] M. Koivisto. Advances in exact Bayesian structure discovery in Bayesian networks. In *UAI*, 2006.

[KS04] M. Koivisto and K. Sood. Exact Bayesian structure discovery in Bayesian networks. *J. of Machine Learning Research*, 5:549–573, 2004.

[MGR05] F. Markowetz, S. Grossmann, and R.Spang. Probabilistic soft interventions in Conditional Gaussian networks. In *10th AI/Stats*, 2005.

[MKTG06] V. Mansinghka, C. Kemp, J. Tenenbaum, and T. Griffiths. Structured priors for structure learning. In *UAI*, 2006.

[ML98] Andrew W. Moore and Mary S. Lee. Cached sufficient statistics for efficient machine learning with large datasets. *J. of AI Research*, 8:67–91, 1998.

[MS03] F. Markowetz and R. Spang. Evaluating the effect of perturbations in reconstructing network topologies. In *Proc. 3rd Intl. Wk. on Distrib. Stat. Computing*, 2003.

[Mur01] K. Murphy. Active learning of causal Bayes net structure. Technical report, Comp. Sci. Div., UC Berkeley, 2001.

[MY95] D. Madigan and J. York. Bayesian graphical models for discrete data. *Intl. Statistical Review*, 63:215–232, 1995.

[Pea00] J. Pearl. *Causality: Models, Reasoning and Inference*. Cambridge Univ. Press, 2000.

[Rob73] R. W. Robinson. Counting labeled acyclic digraphs. In F. Harary, editor, *New Directions in the Theory of Graphs*, pages 239–273. Academic Press, 1973.

[SGS00] P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search*. MIT Press, 2000. 2nd edition.

[SPP+05] 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.

[TK01] S. Tong and D. Koller. Active learning for structure in Bayesian networks. In *Intl. Joint Conf. on AI*, 2001.

[TP01a] J. Tian and J. Pearl. Causal discovery from changes. In *UAI*, 2001.

[TP01b] J. Tian and J. Pearl. Causal discovery from changes: a Bayesian approach. Technical report, UCLA, 2001.

[WGH06] A. Werhli, M. Grzegorczyk, and D. Husmeier. Comparative evaluation of reverse engineering gene regulatory networks with relevance networks, graphical Gaussian models and Bayesian networks. *Bioinformatics*, 22(20):2523–2531, 2006.