# Causal discovery through MAP selection of stratified chain even graphs

Robert G. Cowell
Faculty of Actuarial Science and Insurance
Cass Business School
City University London
106 Bunhill Row
London EC1Y 8TZ
UK.

James Q. Smith
Department of Statistics
University of Warwick
Coventry CV4 7AL
UK

September 25, 2013

**Abstract**

We introduce a subclass of chain event graphs that we call *stratified chain event graphs*, and present a dynamic programming algorithm for the optimal selection of such chain event graphs that maximizes a decomposable score derived from a complete independent sample. We apply the algorithm to a such a dataset, with a view to deducing the causal structure of the variables. We show that the algorithm is suitable for small problems. Similarities with and differences to a dynamic programming algorithm for MAP learning of Bayesian networks are highlighted, as are the relations to causal discovery using Bayesian networks.

**Keywords**

Causality; chain event graph; event tree; stratified chain event graph; staged event tree; structural learning; MAP estimation.

# 1 Introduction

For a long time it has been recognised that whilst being a highly useful modelling tool, a discrete Bayesian Network (BN) does not necessarily depict all the symmetries in a problem. Furthermore, as more applications were explored, it was discovered that many of the conditional probability tables associated with them were very sparse because the natural description of the underlying problem had many logical constraints, and under

some situations variables were either degenerate or, worse, had no meaning. This was particularly true when a model described how events might unfold (Shafer, 1996).

To address the first problem context specific BNs began to be developed e.g. (McAllester et al., 2004), (Boutilier et al., 1996), followed by object orientated BNs (Koller and Pfeffer, 1997) which later formed the basis of the HUGIN architecture (Hugin Expert A/S, 2012). Another stream of research, often motivated by table sparsity, proposed representing subgraphs of adjacent vertices in BNs as probability trees (Salmerón et al., 2000). In fact it was noted – long ago – by Friedman and Goldszmidt (1998) that it was possible to score such tree constrained context specific BNs in closed form.

Probability Decision Graphs (PDGs) (Wallace and Patrick, 1993) provided an alternative class of graphical models quite distinct from the BN (Jaeger, 2004) in both their substance and the semantics of their topology. The class of Chain Event Graphs (CEGs) (Smith and Anderson, 2008) is a class of tree based models which contain both the class of PDGs and all context specific BNs as special cases. This coloured graph can be used as an efficient tool of propagation (Thwaites et al., 2008) and causal representation (Thwaites et al., 2010; Thwaites, 2013) and analysis. It is extremely flexible and has recently been shown how, for problems with small numbers of variables, closed form selection can fully depict and identify previously undiscovered structure, (Freeman and Smith, 2011a,b; Smith and Freeman, 2011; Barclay et al., 2012) in sometimes very difficult and heterogeneous environments.

There are however two snags associated with this expressive class of models. The first is that, representationally, the graphs are usually larger than a BN which can make them less transparent. This problem can often be partially overcome by zooming down into areas of the graph of interest in the model in moderately sized problems. A more serious challenge is that models whose probability space has even a moderate number of atoms is absolutely gigantic and dwarfs BN model space by orders of magnitude. The size of this space presents particular challenges to model selection.

In this paper we begin to address this problem. We are concerned with the optimal selection of a CEG structure after observing a complete random sample from a given population. Model selection and learning of CEGs from data has already been considered in the literature (Thwaites et al., 2009; Freeman and Smith, 2011b,a). Barclay et al. (2013a) used a two-stage procedure for selecting a CEG for a dataset concerning social and family factors on children's health in a New Zealand birth cohort (Fergusson et al., 1986). In the first stage a few high scoring Bayesian networks were found by an exhaustive search. These were then embellished to form more general CEGs that had a significantly better score than the Bayesian networks from which they were derived.

In this paper we introduce a restricted class of CEGs that we call the *stratified chain event graph* (SCEG). These were used implicitly in the model search of Barclay et al. (2013a) and are especially useful for expressing standard types of causal hypotheses. We exploit the close relationship of BNs to SCEGs to develop an algorithm for their selection that is similar to the dynamic programming method of MAP selection of BNs presented by Silander and Myllymäkki (2006). That is, we show how dynamic programming can be applied to SCEG model selection, which contains as a special case BN model selection.

This allows us to search for putative causal hypotheses expressed as SCEGs from observational studies. These suggestive hypotheses can then be experimentally tested through appropriately designed experiments. We demonstrate through an example how this can allow us to discover more diverse and subtle causal hypotheses than if we were to simply use a BN approach.

The plan of the paper is as follows. We begin with a review of terminology for event trees and CEGs, show how to find the BDEu score of a CEG using a complete sample of independent observations, and introduce SCEGs. We illustrate by examples the representation of BNs by staged event trees. We discuss causal representation using CEGs before presenting algorithms for learning BNs and SCEGs. We first present learning when a node ordering is given. This is followed by a review of the the dynamic programming (DP) approach for learning BNs and a novel generalization of the DP algorithm for SCEGs. As a by-product, this shows that our DP procedure contains BN learning as a special case. We apply the algorithm in Section 6 to the dataset examined by Barclay et al. (2013a) and then close with a general discussion of issues raised.

# 2 Event trees and stages

## 2.1 Representation using Staged Trees and CEGs

Probability trees (Shafer, 1996), their control analogues - decision trees - and their embellishments - coalescent trees, (Olmsted, 1983; Bielza and Shenoy, 1999) and probability decision graphs (Oliver, 1993; Jaeger, 2004), have been found to be a very natural and expressive framework for probability and decision problems and provide an excellent framework for describing sample space asymmetry and inhomogeneity in a given context (see e.g. French and Insua (2000)). Let an event tree $\mathcal{T}$ have vertex set denoted by $V(\mathcal{T})$ and (directed) edge set denoted by $E(\mathcal{T})$. Henceforth call its non-leaf vertices *situations* and denote the set of situations by $S(\mathcal{T}) \subset V(\mathcal{T})$. To incorporate a notion of independence on to a CEG it is necessary to supplement the tree with a partition of its situations, or more precisely its associated florets. This partition can be expressed through, where necessary, allocating some of the vertices in $S(\mathcal{T})$ and their emanating edges in $E(\mathcal{T})$ a colour.

Call the subtree $\mathcal{F}(v)$, $v \in S(\mathcal{T})$ of $\mathcal{T}$ a *floret* when the vertex set of $\mathcal{F}(v)$ is $v$ and all its children and the edge set of $\mathcal{F}(v)$ consists of all the edges between $v$ and its children. Clearly an event tree can be decomposed into its florets and a rule identifying which children of parents in $S(\mathcal{T})$ are the parent or root in an adjacent floret. Now partition the set of florets $\{\mathcal{F}(v) : v \in S(\mathcal{T})\}$ into clusters called *floret clusters*. A useful partition for expressing various forms of conditional independence is one that places two florets $\mathcal{F}(v_1), \mathcal{F}(v_2)$ $v_1, v_2 \in S(\mathcal{T})$ in the same cluster if the model asserts that there is a mapping between the edges in $\mathcal{F}(v_1)$, and the edges in $\mathcal{F}(v_2)$ such that the edge probabilities inherited from $\mathcal{T}$ identified by this map are the same. The two florets and their root situations $v_1, v_2 \in S(\mathcal{T})$ are then said to be in the same *stage*.

Whenever $\mathcal{F}(v_1)$ and $\mathcal{F}(v_2)$ are in the same stage we allocate their root vertices $v_1$

and $v_2$ in $\mathcal{T}$, together with their emanating edges, the same colour. The emanating edges are also coloured to identify which edges are identified in the map described above. So when two edges in $E(\mathcal{T})$ emanating from different vertices are allocated the same colour then their associated edge probabilities are believed to be identical. The resulting coloured graph is called a *staged tree*. Note that in an event tree with no additional structure the clusters are simply the singleton situations of $\mathcal{T}$.

A more concise graphical representation of the staged tree partition on a staged tree is finer than the one into stages and is called the *position partition*. Let $\mathcal{T}(v)$ denote the coloured subtree of $\mathcal{T}$ rooted at $v$. Two situations $v, v'$ in the same stage are also in the same position if there is an isomorphic map mapping between the two coloured subtrees $\mathcal{T}(v) \to \mathcal{T}(v')$ : so in particular not only the two subtrees are identical but also the colours of any edges in the map between the two trees correspond. Thus for example when $v, v'$ are only a distance 1 from a leaf node then $\mathcal{T}(v) = \mathcal{F}(v)$ and $\mathcal{T}(v') = \mathcal{F}(v')$ and so they will be in the same position if they are in the same stage. But if they are each a distance 2 from a leaf node of the tree, not only do we need these two situations to be in the same stage but also their children.

The chain event graph (CEG) $\mathcal{C}(\mathcal{T})$ giving a concise representation of a staged tree $\mathcal{T}$ is defined in (Smith and Anderson, 2008; Thwaites et al., 2010; Thwaites and Smith, 2011). This allows the structure of the staged tree to be fed back to a client in a more transparent way. Briefly its vertex set consists of the set the positions of $\mathcal{T}$ together with a sink vertex we label by $w_\infty$. There is a directed edge from position $w \to w'$ for every situation $v \in w$ leading to a situation $v' \in w'$, or from $w \to w_\infty$ for every situation $v \in w$ leading to a leaf vertex of $\mathcal{T}$. Colours on both vertices and edges are inherited from the corresponding edges in the staged tree if and only if the colour will appear more than once on the new graph.

## 2.2 Model Selection over CEGs

Conjugate learning for CEGs which can accommodate not only sampling schemes but also causal experimental data (e.g. like (Cooper and Herskovits, 1992)) is now well documented (Thwaites et al., 2009; Smith, 2010; Smith and Anderson, 2008; Freeman and Smith, 2011b). These methods and the formulae closely resemble analogous learning in discrete Bayesian Networks under full sampling of the net (Heckerman, 1998) and those BNs with internal trees structures, (Friedman and Goldszmidt, 1998). Briefly, suppose a CEG $\mathcal{C}$ has stages $\{u_i : 1 \leq i \leq k\}$ and each situation in $u_i$ has $k_i$ emanating edges labelled $(e_{i1}, e_{i2}, \ldots, e_{ik_i})$ with associated probability vector $\boldsymbol{\pi}_i = (\pi_{i1}, \pi_{i2}, \ldots, \pi_{ik_i})$ where $\sum_{j=1}^{k_i} \pi_{ij} = 1$, $\pi_{ij} > 0$, $1 \leq j \leq k_i, 1 \leq i \leq k$. Its likelihood $L(\boldsymbol{\pi})$ then takes the separable form

$$L(\boldsymbol{\pi}) = \prod_{i=1}^{k} L_i(\boldsymbol{\pi}_i)$$

where $\boldsymbol{\pi} = (\boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \ldots, \boldsymbol{\pi}_k)$. Here, for $1 \leq i \leq k$,

$$L_i(\boldsymbol{\pi}_i) = \prod_{j=1}^{k_i} \pi_{ij}^{n_{ij}}$$

where $n_{ij}$ denotes the number of units in the sample that arrive at stage $u_i$ and the pass along the $j^{th}$ edge of the associated floret.

Note that, as for the BN, if a priori the vectors of stage probabilities $\boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \ldots, \boldsymbol{\pi}_k$ of $\boldsymbol{\pi}$ are all mutually independent, then they will also be independent a posteriori. Furthermore if $\boldsymbol{\pi}_i$ has a Dirichlet distribution $Di(\boldsymbol{\alpha}_i)$ where $\boldsymbol{\alpha}_i = (\alpha_{i1}, \alpha_{i2}, \ldots, \alpha_{ik_i})$ a priori where $\sum_{j=1}^{k_i} \pi_{ij} = 1$, $\pi_{ij} > 0$, $1 \leq j \leq k_i$, then

$$p(\boldsymbol{\pi}_i) = \frac{\Gamma(\alpha_{i1} + \ldots + \alpha_{ik_i})}{\Gamma(\alpha_{i1}) \ldots \Gamma(\alpha_{ik_i})} \prod_{j=1}^{k_i} \pi_{ij}^{\alpha_{ij}}$$

and it is straightforward to verify that $\boldsymbol{\pi}_i$ has a Dirichlet distribution $Di(\boldsymbol{\alpha}_i^*)$ a posteriori where $\alpha_{ij}^* = \alpha_{ij} + n_{ij}$, $1 \leq j \leq k_i, 1 \leq i \leq k$.

Each model can now be scored using a variety of methods. Although not critical to this development in this paper we have chosen to illustrate our methodology using using the popular log marginal likelihood score $Q(\mathcal{C})$ of each CEG $\mathcal{C}$.

This can be calculated by the formula

$$Q(\mathcal{C}) = \sum_{i=1}^{k} \left\{ \sum_{j=1}^{k_i} \left( \log \Gamma(\alpha_{ij}^*) - \log \Gamma(\alpha_{ij}) \right) - \left( \log \Gamma(\overline{\alpha}_i) - \log \Gamma(\overline{\alpha}_i^*) \right) \right\} \tag{2.1}$$

where $\overline{\alpha}_i \triangleq \alpha_{i1} + \ldots + \alpha_{ik_i}$ and $\overline{\alpha}_i^* \triangleq \alpha_{i1}^* + \ldots + \alpha_{ik_i}^*$, $i = 1, 2, \ldots, k$. The MAP CEG model is then the one that maximizes this function and so chooses that CEG model which given the observed sample is a posteriori most probable. Note that it is the linearity of this score function which makes possible the search algorithm of this paper.

To employ this search method it is first necessary to input appropriate prior densities of the hyper-parameters $\alpha_i : i = 1, \ldots, k$, and a prior over each candidate model. Freeman and Smith (2011a) described a characterization of priors analogous to the one given in (Geiger and Heckerman, 1997) and (Heckerman, 1998), so that models with the same substructures in their description have the same local priors on these shared features. This allows us to choose priors over each model in the class of CEGs sharing the same tree compatibly with those of the saturated tree where stages are simply situations. This characterization in (Freeman and Smith, 2011a) implies that the prior of edge probabilities on the floret of each stage on any possible CEG has to have a Dirichlet distribution. Furthermore to ensure compatibility across different candidate CEGs we need to set these hyperparameters to be proportional to the power $\overline{\alpha}$ of a likelihood $L(\boldsymbol{\pi})$ of the monomial form above. Here we use the simplest default vague setting for this phantom likelihood and assume that the phantom sample giving rise to this likelihood consists of observing a set of units where exactly one

unit passes from the root to each leaf of the tree. This gives exchangeable beliefs over the probabilities of the atoms of the space. The parameter $\overline{\alpha}k$ can be then thought of an effective sample size parameter for each CEG with the given tree. The simplest choice sets $\overline{\alpha} = 1$ - see also (Freeman and Smith, 2011b), (Barclay et al., 2013a),(Barclay et al., 2012). For the saturated model this assigns a uniform prior over the leaves of the tree. Of course depending on the context, other choices of prior might be more appropriate. Having set $\boldsymbol{\alpha}_i$ $i = 1, 2, \ldots, k$ in this way for the saturated tree with trivial stage structure, then the modularity properties determine the prior $\alpha$ hyperparameter associated with the edge of a floret of each stage of all other competing CEGs as well. This edge probability is simply $\overline{\alpha}$ times the number of times a unit passes along this edge in the phantom sample described above (Freeman and Smith, 2011a). All stages within a particular CEG are then assumed to be independent of each other. A full description of this process and its justification can be found in (Freeman and Smith, 2011a). This then defines the full prior specification of the models over the class.

## 2.3 Stratified chain event graphs

The general space of all CEGs given in Smith and Anderson (2008) is huge. Furthermore its associated statistical models are not all compatible with a search across the very specific *types* of causal hypotheses which specify relationships between a predetermined set of measurement variables. So for reasons of compatibility with analogous Bayesian Network causal discovery algorithms, in this paper we restrict our search so that it applies in a similar setting. Thus suppose we have given to us a vector $\boldsymbol{X} = \{X_1, X_2, \ldots, X_n\}$ of $n$ preferred variables where $X_j$ takes values $x_j \in \mathcal{X}_j$. We want to investigate various putative causal hypotheses concerning the components of $\boldsymbol{X}$. Let $I$ denote a permutation of $\{1, 2, \ldots, n\} \longmapsto \{i_1, i_2, \ldots, i_n\}$ which is used to reorder the components of $\boldsymbol{X} \longmapsto \{X_{i_1}, X_{i_2}, \ldots, X_{i_n}\} \triangleq \boldsymbol{X}(I)$ and let $\mathcal{X}^{(k)}(I) = \mathcal{X}_{i_1} \times \mathcal{X}_{i_2} \times \cdots \times \mathcal{X}_{i_k}$

### Definition

Say that an event tree $\mathcal{T}(\boldsymbol{X}, I)$ is $\boldsymbol{X}-$ *compatible* if its vertex set $V(\mathcal{T}(\boldsymbol{X}, I))$ consists of a root vertex $v_0$ together with a set of vertices $v(\boldsymbol{x}^{(k)})$, one for each $\boldsymbol{x}^{(k)} = (x_{i_1}, x_{i_2}, \ldots, x_{i_k})$, and $1 \leq k \leq n$.

Note that under this indexing each of the non-root vertices $v(\boldsymbol{x}^{(k)}) \in V(\mathcal{T}(\boldsymbol{X}, I))$ , $\boldsymbol{x}^{(k)} \in \mathcal{X}^{(k)}(I)$ is the same distance from the root $v_0$. Also note that any edge emanating from $v(\boldsymbol{x}^{(k)})$ can be labelled by a value $x_{i_{k+1}} \in \mathcal{X}_{i_{k+1}}$ of $X_{i_{k+1}}$: i.e. a possible value of the next variable on the list of components determined by $I$.

We now define a subclass of CEGs which are particularly straightforward to relate to causal hypotheses about $\boldsymbol{X}$ in a way analogous to those used in the Causal Discovery algorithms for BNs proposed by various authors: see e.g. (Spirtes et al., 1993; Pearl, 2009).

**Definition**

An $\boldsymbol{X}$ - *stratified chain event graph* ($\boldsymbol{X}$ - SCEG) has an associated $\boldsymbol{X}$ - compatible event tree $\mathcal{T}(\boldsymbol{X}, I)$ for some permutation $I$. Its stage partition has the following properties

1. The stage partition consists of the root node of $\mathcal{T}(\boldsymbol{X}, I)$ and subsets of the form

$$u(B^{(k)}) \triangleq \left\{v(\boldsymbol{x}^{(k)}) : \boldsymbol{x}^{(k)} \in B^{(k)} \subseteq \mathcal{X}^{(k)}(I)\right\}$$

   for $1 \leq k \leq n-1$.

2. The mapping associating two florets $\mathcal{F}(v_1(\boldsymbol{x}^{(k)}))$ and $\mathcal{F}(v_2(\boldsymbol{x}^{(k)}))$ in the same stage floret cluster always maps the edges so that their labels $x_{i_{k+1}} \in \mathcal{X}_{i_{k+1}}$ on the full tree coincide

The first condition restricts the class of CEGs we search to ones where floret clusters can only contain root vertices that are the same distance from the root. In particular therefore these are associated with values of a particular subvector of $\boldsymbol{X}$. Of course, this is far from the only type of CEG on which causal relations can be defined - see for example (Thwaites et al., 2010; Thwaites, 2013). However it is one where correspondences between causal hypotheses in Bayesian Networks and context specific conditional independences can most easily be made.

The second condition simply demands that when two florets $\mathcal{F}(v_1(\boldsymbol{x}^{(k)}))$ and $\mathcal{F}(v_2(\boldsymbol{x}^{(k)}))$ are in the same floret cluster, the conditional distribution of the next list variable $X_{i_{k+1}}$ given $v(\boldsymbol{x}_1^{(k)})$ occurs is the same as when $v(\boldsymbol{x}_2^{(k)})$ occurs (rather than some $1-1$ function of $X_{i_{k+1}}$ being the same). This is a natural assumption to make about all the types of example we consider in this paper. Note that the subclass of $\boldsymbol{X}$ - SCEGs is still very large and contains as a subset all context specific BNs on the variables $\boldsymbol{X}$.

# 3 Representing BNs by staged event trees

We now present a small example to illustrate the connection between a BN and a staged event tree. Suppose that we have three binary random variables, $A$, $B$ and $C$. Recall that an event tree has an ordering of variables, so we consider all possible Bayesian networks in the three variables which have the topological ordering $A$ followed by $B$ and then $C$, which we denote by $A \prec B \prec C$. There are eight such networks shown in Figure 1.

Figure 2 shows an event tree for the three variables. In this figure the probabilities associated with the edges coming out of node $B_1$ correspond to the conditional probability distribution $P(B|A=1)$ those out of $C_{01}$ to $P(C|A=0, B=1)$, and so on. (Note that the numbers on the edges are the states of the variable, not the probabilities.)

Now consider for example, Bayesian network (i) in Figure 1. In this network we have that the three variables are mutually independent. This independence is represented by nodes $B_0$ and $B_1$ being in the same stage, and the four nodes $C_{00}$, $C_{01}$, $C_{10}$, and $C_{11}$ also all being in another stage. Hence the BN induces the following partition of the nodes of the

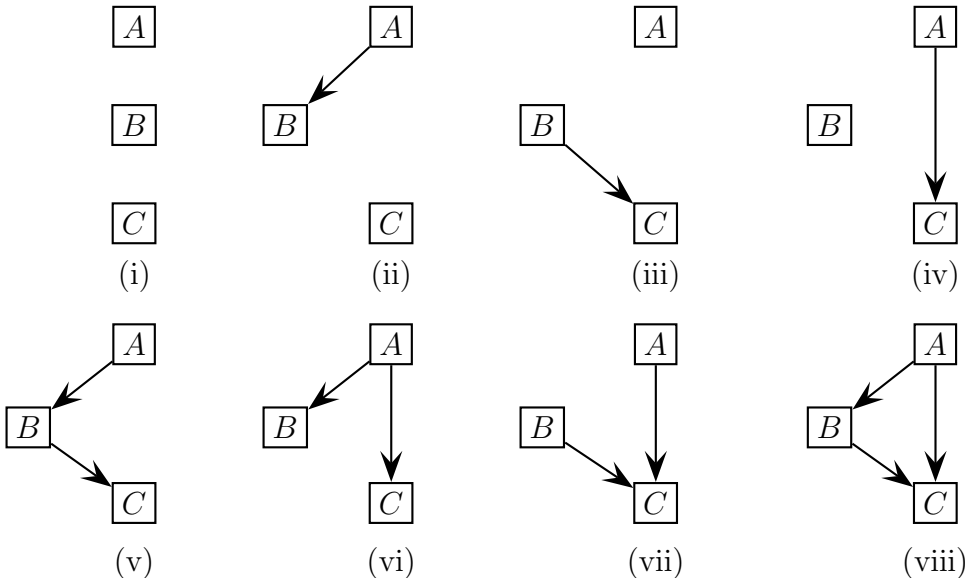


Figure 1: The eight Bayesian networks in the three binary random variables $A$, $B$ and $C$ having the topological ordering $A \prec B \prec C$.

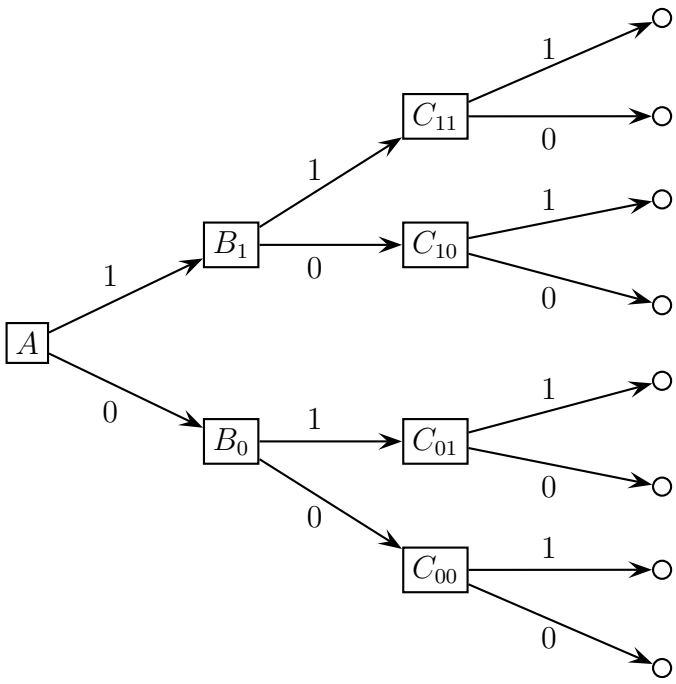


Figure 2: An event tree for the three binary random variables $A$, $B$ and $C$ in the order $A \prec B \prec C$. Note that the edge labels denotes states, not probabilities.

Table 1: Partition of nodes in the event tree into stages by BNs

| BN | Stage partition | | |
|---|---|---|---|
| (i) | $(A)$ | $(B_0, B_1)$ | $(C_{00}, C_{01}, C_{10}, C_{11})$ |
| (ii) | $(A)$ | $(B_0),\ (B_1)$ | $(C_{00}, C_{10}, C_{01}, C_{11})$ |
| (iii) | $(A)$ | $(B_0, B_1)$ | $(C_{00}, C_{10}),\ (C_{01}, C_{11})$ |
| (iv) | $(A)$ | $(B_0, B_1)$ | $(C_{00}, C_{01}),\ (C_{10}, C_{11})$ |
| (v) | $(A)$ | $(B_0),\ (B_1)$ | $(C_{00}, C_{10}),\ (C_{01}, C_{11})$ |
| (vi) | $(A)$ | $(B_0),\ (B_1)$ | $(C_{00}, C_{01}),\ (C_{10}, C_{11})$ |
| (vii) | $(A)$ | $(B_0, B_1)$ | $(C_{00}),\ (C_{01}),\ (C_{10}),\ (C_{11})$ |
| (viii) | $(A)$ | $(B_0),\ (B_1)$ | $(C_{00}),\ (C_{01}),\ (C_{10}),\ (C_{11})$ |

event tree into stages: $(A), (B_0, B_1), (C_{00}, C_{01}, C_{10}, C_{11})$. In a similar manner the other BNs also induce partitions of the event tree nodes into stages, a complete list is given in Table 1. It is here that we can see how CEGs generalize BNs: the conditional independences explicit in the structure of a BN induces a *restricted* set of partitions over the nodes of the event tree into stages. Note that by restricting attention to partitions in each of the (vertical, as drawn) levels of the event tree, these CEGs are instances of SCEGs. A SCEG also induces a set of partitions, but is not limited to the restricted set required for a BN on the original variables. For example, the stage partition $(A), (B_0, B_1), (C_{00}, C_{01}, C_{10}), (C_{11})$ defines a valid SCEG in which the outcome of $C$ is independent of both $A$ and $B$ unless both these variables take the value 1. This context specific independence is not explicitly representable graphically by a standard BN, but is shown as a SCEG in Figure (3).

# 4 Graphical Causal Discovery and CEGs

There are now various well developed exploratory causal discovery procedures for BNs whose vertices are the random variables $(X_1, X_2, \ldots, X_n)$. Typically these methods search over the space of BNs to find the best explanatory model over the observed population either by scoring - e.g. using a Bayes Factor scoring method (Heckerman (1998); Cussens (2011)) (as we have used here for CEGs) - or sequentially testing for conditional independences (eg Spirtes et al. (1993)). Once the best explanatory BN has been discovered the topology of the BN is inspected. Any BN lies in an equivalence class of Markov equivalent graphs. All elements in this equivalence class have the same *skeleton* - i.e. the same undirected graph where all directed edges are replaced by undirected ones.

Scientific interest often needs to discover the direction of "causal" relationships rather than dependence relationships. A heroic assumption is that any edge direction between two variables in the best explanatory model which is common to all BNs in its equivalence class might be hypothesised to be such a causal link. This argument is based on the hypothesis that the data is generated by some BN. Pearl (2009) argues that the only conditional independences that we can expect to discover will be those logically entailed by this BN since

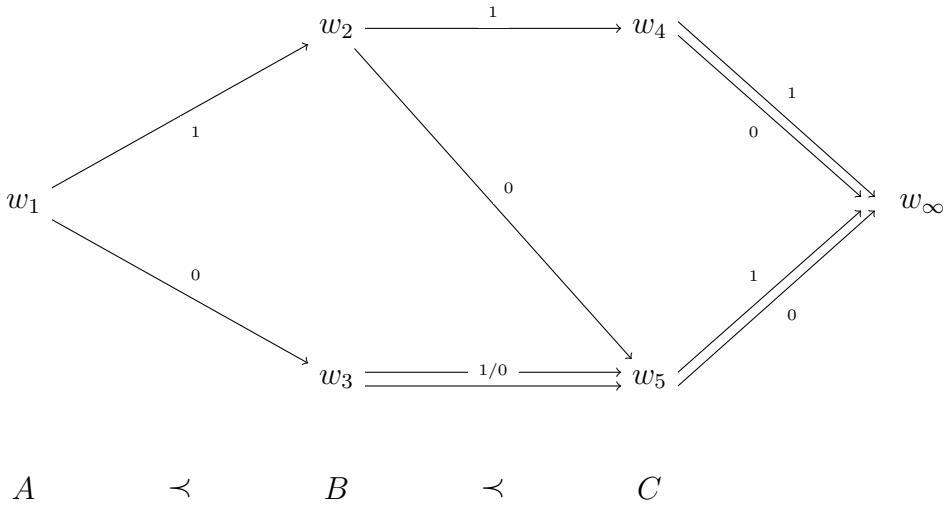


Figure 3: SCEG corresponding to the stages partition (iv) in Table 1. Node $w_1$ corresponds to the stage partition $(A)$, node $w_2$ to the stage $(B_1)$ and $w_3$ to $(B_0)$. Node $w_4$ corresponds to the stage partition $(C_{11})$, and $w_5$ to the stage partition $(C_{00}, C_{01}, C_{10})$.

the probability that additional conditional independences manifest themselves in an infinite population should effectively be zero. So in this sense with a sufficiently large sample the best explanatory model contains all those and only those conditional independences that exist.

Given any BN those edges which have an unambiguous direction across all other Markov equivalent graphs are known - these are the directed edges in its *essential* graph (Andersson et al., 1996), defined by its *pattern* (Verma and Pearl, 1991; Pearl, 2009) - and can be quickly identified. Note that Pearl argues - assuming what is observed is actually the margin of a much larger BN - that only a subsets of these edge directions can possibly be putative causes.

Of course all the assumptions above are extremely speculative and the argument linking causality and conditional independence in this way is debatable. On the other hand these search methods can provide a very powerful exploratory data tool for selecting experiments to establish causal relationships that might subsequently be defined in a more watertight sense. Thus an atomic causal intervention is an experiment which artificially sets one of the values of the variables to a selected value. Pearl and others argue as follows. For the BN to be causal the effect of this intervention compared with the unintervened system would be to leave the distribution of all non- descendent variables unchanged but change the distributions of all descendent variables to be consistent with conditioning on the intervened variable. It is often possible to design experiments to decide whether this prediction is supported by data. This process has now been widely applied in many applications (Peer et al., 2001; Sachs et al., 2005; Ramsey et al., 2011; Maathuis et al., 2010).

What we argue here is that - as was briefly eluded to in (Barclay et al., 2013a) - exactly

the same procedure of causal discovery can also be applied to CEGs, using the output of model search algorithms described in this paper. We here accommodate this idea in the proposed search procedure summarized here:

- We find the highest scoring model.
- We identify its Markov equivalence classes over our chosen model space.
- We identify the relationships where the directionality is uniform over this equivalence class.
- We interpret this direction as a putative causal hypothesis embedded within the chosen model.

We illustrate this with some examples of using CEGs for causal representation.

## 4.1 Example 1. A Causal switch mechanism

This is the simplest example of a CEG which is not a BN. Two binary variables $X, Y$ each take values either $-1$ or 1, $X$ taking the value 1 with probability $p$ and $Y$ with probability $q_{-1}$ when $X = -1$ and probability $q_1$ when $X = 1$. Suppose $q_1 = 1 - q_{-1} = q$. Then this is a CEG of the kind in the first two bullets, and corresponds to the staged tree in Figure 4: the model implicitly states that $XY \perp\!\!\!\perp X$. Note that this relationship is not symmetric since it does not imply $XY \perp\!\!\!\perp Y$. This is the simplest "causal" CEG. Here we can say that the value of $X$ determines whether or not $Y$ switches from favouring a value of 1 to favouring a value of $-1$. It can be shown that this is the only type of unambiguous order that can be described by a tree on two binary variables.

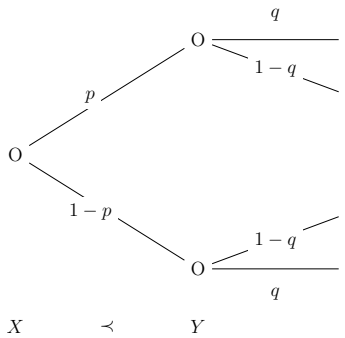


Figure 4: The simplest chain event graph that is not Bayesian network.

## 4.2 Example 2. Causes by extent

A second type of causal CEG is expressed in Figure 5, where $X$ takes three levels and $Y$ two levels. Here the upper two situations are in the same stage associated with $X = 1$ or $X = 2$ are the same - so conditional on these two events the probability distribution of $Y$ is the same. However when $X = 3$ there is a change in the distribution of $Y$. This can be given a causal explanation by saying "$X$ taking its highest value causes $Y$ to change but at its two lower values it does not impact upon $Y$". Again this is the only tree in the space that lies in the equivalence class so in particular $X$ lies before $Y$ in the causal order under this model. Note there are two other such CEGs which would suggest a similar causal order - stages defined by $X = 1$ and $X = 3$ and $X = 2$ and $X = 3$. We cannot discover a causal order of this type that lists $Y$ unambiguously before $X$ because we need more levels in $Y$ or more variables to do this.

This demonstrates that the potential for finding causal orderings is much richer than the search over BNs. If we use the causal orders defined by V- structures which are the directed edges in an essential graph then we need at least three variables for any causation to be discovered whilst the refinement by Pearl (2009) requires at least 4.

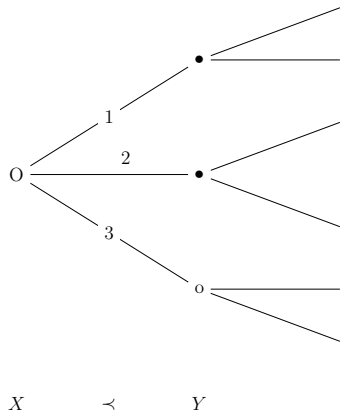


Figure 5: A second chain event graph that is not a Bayesian network.

Of course the leap from a best fitting model to a model that predicts causation under manipulation is just as speculative as it was for the BN. However the effects of a manipulation are just as well defined for a Causal CEG as they are for a Causal BN and described in (Thwaites et al., 2010) and (Thwaites, 2013). So for example in the staged tree in the CEG above a manipulation of $X$ so that it takes a value of 1 or 2 will give rise to an observation from the $Y$ variable having a distribution from the shared stage. However forcing $X$ to take the value 3 would give rise to the an observation $Y$ drawn from the $Y|X = 3$ distribution associated with the unintervened population. On the other hand the causal hypothesis will lead us to conclude that intervening on $Y$ will have no effect on the distribution of $X$. Having generated such hypotheses it is then possible to construct experiments to examine whether or not these hypotheses are actually justified by intervening on a number of units

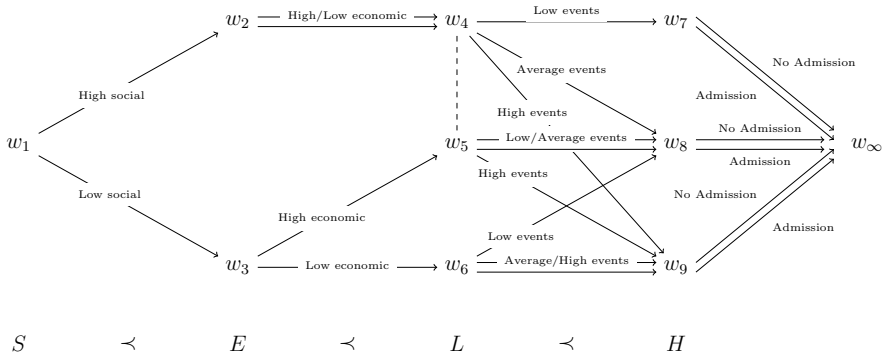


Figure 6: MAP CEG for the CHDS data under the restriction of the node ordering $X_1, X_2, X_3, X_4$.

and observing what happens.

## 4.3 Example 3 Christchurch study: a SCEG expressing a causation hypothesis

This example is based on a subset of data collected for the Christchurch Health and Development Study (CHDS), where many explanatory variables associated with the hospital admissions, $H$ – here classified into three levels (low, medium, high) due to illness or accidents – of 2635 children born in 1977 were tracked. On the basis of this study Fergusson et al. (1986), using an initial factor analysis, demonstrated that in the first five years of their life, the social background of the child's family $S$ (classified into two levels), the family's financial status, $E$ (classified into two levels), and the occurrence of various life events, $L$ (classified into 3 levels (low, medium, high)) such as the death of a close relative or divorce in the family, all had a significant impact on hospital admissions. For more details of the definitions of these categories see study see Barclay et al. (2013a) and Fergusson et al. (1986).

In (Barclay et al., 2013a) it was demonstrated using a greedy search algorithm and a given causal ordering of the explanatory variables, how a CEG could be found that performed much better under a Bayes Factor search above than any BN. Here we have used their data in a full search of the CEG space using our dynamic programming approach to see if there is another CEG that explains the data even better under this MAP score. The CEG they found is shown in Figure 6. It is easy to check that the only Markov equivalent CEGs in terms of the order are ones where the first two variables - the social and the economic variable - are permuted. This actually makes some sense in the sense that these are both simply established covariates of the units in the population so have a similar role in the description of this process. Thus we can conclude for the CEG a causal order of the form $S$,$E \prec L \prec H$. But note that since the search space used here implicitly demands the causal order to not allow a causal relationship to violate the order of variable

entered into the tree - and a greedy search algorithm was used. To genuinely search over the space of all plausible causal explanations of the data ideally we would like to find the genuinely highest scoring CEG associated with each tree and also search over such trees. This however demands us to develop efficient methods to complete this search even for relatively small scale problems like this one. This motivated the development we now describe in this paper.

# 5 Learning with node order

Before considering the structural learning of CEGs, we first consider the related problem of learning the structure of a Bayesian network. The simplest situation is when a node ordering is given, which we examine first and then present a corresponding algorithm for CEGs. We then review the dynamic programming method for learning BNs when a node order is not available. In both cases we assume that we have a set of $n$ discrete random variables $X_1, X_2, \ldots, X_n$ where for each $i \in \{1, \ldots, n\}$ the variable $X_i$ has state space dimension of $k_i$. We also assume a complete sample of $N$ observations, and we shall rank models according to their (decomposable) BDEu score.

## 5.1 BN Structure learning with a given node ordering

This is by far the simplest case. Without loss of generality, we may assume that the variables are ordered by their index numbers, so that $X_1 \prec X_2 \prec \cdots \prec X_n$. Let $\mathcal{G}^{\prec}$ denote the set of Bayesian network structures consistent with the ordering, and consider a graph $g \in G^{\prec}$. For variable $X_i$, let $\mathrm{pa}(i)$ denote the set of parents of $X_i$ in $g$. The overall score of the network $g$ may be written additively as

$$S^g = \sum_i S^g_{i,\mathrm{pa}(i)}$$

where $S^g_{i,\mathrm{pa}(i)}$ is the component of the score associated with variable $i$ having parent set $\mathrm{pa}(i)$, which for the BDEu score is given by

$$S^g_{i,\mathrm{pa}(i)} = \sum_{k=1}^{k_i} \sum_{j=1}^{J} \log \frac{\Gamma(\alpha_{ijk} + n_{ijk})}{\Gamma(\alpha_{ijk})} - \sum_{j=1}^{J} \log \frac{\Gamma(\alpha_{ij} + n_{ij})}{\Gamma(\alpha_{ij})}$$

where $J$ is the size of the state space of the parent set $\mathrm{pa}(i)$ of variable $X_i$, $\alpha_{ijk} = \alpha/(k_i J)$ with $\alpha_{ij} = \alpha/J$, and $n_{ijk}$ are the marginal counts for $X_i$ and its parent set, with $n_{ij} = \sum_{k=1}^{k_i} n_{ijk}$.

To select the network $g$ having the highest score, we may independently maximise each local score $S^g_{i,\mathrm{pa}(i)}$, for $i \in \{1, \ldots, n\}$. For variable $i$ this means evaluating every one of the $2^{i-1}$ scores from the set of subsets of possible parent sets in the variables $X_1, X_2, \ldots, X_{i-1}$, and choosing the parent set that has the highest score, breaking ties arbitrarily if there is no unique maximum. Finding the optimal scoring network thus requires the evaluation of $O(2^n)$ scores.

## 5.2 SCEG structure learning with a given node ordering

Let $\mathcal{E}^{\prec}$ denote the (stageless) event tree generated by the ordering $X_1 \prec X_2 \prec \cdots \prec X_n$, so that $X_1$ is the root, and $X_n$ generates the leaves of the tree. Let $\mathcal{C}^{\prec}$ denote the set of SCEGs that can be specified over $\mathcal{E}^{\prec}$. For a given $c \in \mathcal{C}^{\prec}$ let $l_i$ denote the partition of the nodes at level $i$ (associated with variable $X_i$) into stages. Then the overall score associated with $c$ may be written, by the decomposition, in the form

$$S_c = \sum_i S^c_{i,l_i}$$

where $S^c_{i,l_i}$ denotes the contribution to the score from the florets on level $i$, and itself is additively decomposable over the elements (florets) of the partition $l_i$:

$$S^c_{i,l_i} = \sum_{\lambda \in l_i} S^c_{i,l_i:\lambda}.$$

In terms of the BDEu score, the score on each floret is given by

$$S^c_{i,l_i:\lambda} = \sum_{k=1}^{k_i} \sum_{j=1}^{J} \log \frac{\Gamma(\beta_{ijk} + m_{ijk})}{\Gamma(\beta_{ijk})} - \sum_{j=1}^{J} \log \frac{\Gamma(\beta_{ij} + m_{ij})}{\Gamma(\beta_{ij})}$$

where $\beta_{ijk} = |\lambda|\alpha/k^i J$ with $\beta_{ij} = |\lambda|\alpha/J$ and $|\lambda|$ denote the number of nodes in the event tree $\mathcal{E}^{\prec}$ at level $i$ in the partition element $\lambda \in l_i$ that have been merged to form a stage. In addition $m_{ijk}$ denotes the marginal counts over the $|\lambda|$ nodes in $\mathcal{E}^{\prec}$ making up the stage $\lambda$ of observing $X_i$ in its $k$-th state and its predecessors in the $j$-th state, and $m_{ij} = \sum_k m_{ijk}$.

Maximising this score may be achieved by maximising the score of each level independently, because of the decomposition. Maximising the score of a given level $i$ is achieved by calculating the scores for every possible partition of the nodes in $\mathcal{E}^{\prec}$ at level $i$, and choosing the partition that maximises this score. Let $\chi_i$ denote the size of the joint state space of the preceding variables $X_1, X_2, \ldots, X_{i-1}$. Then at level $i$ there are $\chi_i$ nodes in the event tree $\mathcal{E}^{\prec}$. The number of partitions of these nodes is given by the $\chi_i$-th Bell number $B_{\chi_i}$.

We make here two observations. The first is that learning BNs is equivalent to learning a restricted set of partitions. The second is that learning SCEG structure is enormously more complex computationally than learning BN structure, because there are considerably more partitions to consider. For example, if we have $n = 4$ binary variables, then there would be 17 local scores $S^g_{i,\mathrm{pa}(i)}$ scores to evaluate for selecting a BN with a given node ordering, compared to 4158 scores to evaluate for SCEG learning with the same ordering. (Taking into account repeated sub-partitions that may occur, these scores may be found from 279 distinct scores that need evaluating, and which may be cached to reuse.)

Given the variable ordering $X_{\sigma(1)} \prec X_{\sigma(2)} \prec \cdots \prec X_{\sigma(n)}$, the general algorithm for SCEG learning is given in Algorithm 1.

**Algorithm 1:** Find the highest scoring SCEG with a given variable ordering

**Input**: A ordered set of variables $X_{\sigma(1)} \prec X_{\sigma(2)} \prec \cdots \prec X_{\sigma(n)}$, and an event tree $\mathcal{E}^{\prec}$ constructed from these.

**Output**: A best scoring SCEG.

1 **for** $i \leftarrow 1$ ***to*** $n$ **do**

2 Find the partition $l_i$ of nodes on level $i$ of $\mathcal{E}^{\prec}$ that maximizes the local score $S^c_{i,l_i}$

3 **return** *The set of partitions* $\{l_i : i = 1..n\}$ *defining the SCEG*

## 5.3 BN Structure learning by dynamic programming

When no node ordering restriction is given, learning a BN structure from data is a much harder problem. One inefficient method is to find the optimal network for each of the $n!$ possible orderings of the variables, and then choose the highest scoring network. The complexity will thus be $O(n!2^n)$, and so this is unfeasible if $n$ is large.

Following on from work by Koivisto and Sood (2004), a Bayesian network structure learning algorithm capable of searching the complete space of Bayesian networks for up to $n = 25$ variables was proposed by Singh and Moore (2005). Subsequently a simpler and more efficient algorithm was proposed Silander and Myllymäkki (2006) with a complexity of $O(n2^n)$ that does an exhaustive search over all possible BNs on $n$ variables, and is able construct maximum scoring Bayesian networks with up to approximately 30 variables.

The key observation, also used by (Singh and Moore, 2005), is that in a directed acyclic graph there is at least one node, called a *terminal node* or *sink*, that does not have any outgoing edges. Removing this sink node results a directed acyclic graph that also has a sink node. Hence the decomposable score is the sum of the score of the sink node plus the score of the network in the remaining $n-1$ nodes. So to find the optimal score, we find the best combination of best sink score plus best score of the remaining variables. This leads to a recursive dynamical programming approach for finding the optimal network, shown in Algorithm 2.

**Algorithm 2:** Find the best scoring Bayesian network

**Input**: A complete dataset on a set of $n$ finite discrete random variables.

**Output**: A best scoring BN.

1 Find the set of possible parent configurations for every variable $X_i$ and their scores

2 Find the best sinks for all $2^n$ subsets of $X$

3 Find a best ordering of best sinks

4 Recover the BN using the sink ordering and the best parents of each sink

5 **return** *Best scoring BN*

The core efficiency of Algorithm 2 is in using an appropriate order for Steps 2 and 3. What we require for the algorithm to work efficiently is to avoid recalculating the sub-network scores that have already been calculated. One possible ordering that achieves this

is to look at the subsets in an order of non-decreasing size. An alternative possibility, given by Silander and Myllymäkki (2006), is to use a lexicographic order of bit vectors that implement the subsets.

## 5.4 SCEG structure learning by dynamic programming

In analogy with BN learning when no node ordering is given, one could consider all possible orderings for generating the event trees, and for each such tree find the partition at each level that maximises the score. The complexity will clearly be $O(n!)$ greater than the complexity of the SCEG learning algorithm presented in § 5.2. However, given that algorithm is practical only for small values of $n$, the overhead will not be so great for $n$ small. Nevertheless, a more efficient algorithm based on dynamic program analogous to that for learning BNs is possible, which we now present.

The key observation is that if an event tree on $n$ ordered variables with $n$ levels has the set of nodes on the final level removed, the result is an event tree with $n-1$ ordered variables on $n-1$ levels. Thus the best scoring SCEG on $n$ variables will be the one that maximizes the score for the $n$-level partition plus the best scoring SCEG for the remaining $n-1$ variables. This allows for the construction of a recursive dynamic programming algorithm to find the highest scoring SCEG, similar to that for learning BNs.

To show the close analogy with the dynamic programming algorithm for BN learning, we shall call the variable associated with the terminal nodes of an event tree on $k$ variables the *sink* variable. Unlike the first step of Algorithm 2 given in Section 5.3, we shall not pre-compute all local scores for reasons explained below, instead we shall compute them as required and cache them. The basic algorithm for learning SCEGs is shown in Algorithm 3.

**Algorithm 3:** Find the best scoring SCEG when no variable ordering is specified

**Input**: A complete dataset on a set of $n$ finite discrete random variables.
**Output**: A best scoring SCEG.
1 Find the best sinks for all $2^n$ subsets of $X$
2 Find a best ordering of best sinks
3 Find the best SCEG using the best ordering
4 **return** *Best scoring SCEG*

We shall now elaborate on each of these steps.

### Step 1: Find the best sink variables

The first step is the most computationally intensive part of the algorithm. We shall do it by looking at subsets of $X$ ordered by increasing size, starting with singleton subsets.

We use two arrays, *scores* and *sinks*, each of size $2^n$, with each element corresponding to a subset of $X$. The algorithm proceeds by examining the subsets of $X$ in order of increasing size, starting with singleton subsets. In addition, there is a function, by $BLS(i, W)$ where

$W \subseteq X \setminus i$, that computes and returns the local score of a tree formed from the set of variables $\{W \cup i\}$ with $i$ the terminal variable associated with the final level of the tree. The algorithm, shown in Algorithm 4, also uses a local variable *skore*.

**Algorithm 4:** Find the best sink variables for every non-empty subset of $X$.

**Input**: A set $X = \{X_1, X_2, \ldots, X_n\}$ of $n$ finite discrete random variables, and a complete dataset of observations for calculating local scores.

**Output**: A set-indexed array `sinks[]` that for each subset $W \subset X$ returns the sink variable for the highest scoring CEG made from the variables of $W$.

```
for i ← 1 to n do
    for W ⊂ X such that |W| = i do
        scores[W] ← 0.0
        sinks[W] ← −1
        for y ∈ W do
            U ← W \ {y}
            skore ← BLS(y, U) + scores[U]
            if sinks[W] = −1 or skore > scores[W] then
                scores[W] ← skore
                sinks[W] ← y
return sinks[]
```

Note that the Best Local Score $BLS(y, U)$ requires constructing an event tree from the variables $U \cup y$ with $y$ being the sink variable. The score itself will not depend on the ordering of the variables $U$ in the tree, however the partition will. We will return to this point later.

### Step 2: Find a best ordering of the best sinks

Having found the best sink variable for every non-empty subset of $X$, finding the best ordering of best sink variables is straightforward. The algorithm first finds the best sink $i$ for the set of all variables $X$, then the best sink for $X \setminus \{i\}$, etc. The algorithm is essential the same as in Silander and Myllymäkki (2006), and uses an $n$ dimensional integer indexed array $ord[\,]$ of variables.

At the end of the algorithm, the array $ord[]$ contains an ordering of the variables of $X$ for the highest scoring SCEG, with $ord[1]$ being the root variable, and $ord[n]$ the terminal variable. The complexity is clearly linear in $n$.

### Step 3: Recover the highest scoring SCEG

Having found the optimal ordering of the variables, we may apply Algorithm 1 to recover the highest scoring SCEG.

**Algorithm 5:** Find the best variable ordering

**Input**: The set indexed array `sinks[]`.

**Output**: A integer-indexed array of the variable ordering for the best scoring CEG.

```
1 left = {X}
2 for i ← n to 1 do
3 |   ord[i] ← sinks[left]
4 |   left ← left \ {ord[i]}
5 return ord[ ]
```

This is now an appropriate place to explain why, in contrast to the DP algorithm for finding the highest scoring BN, we do not pre-compute the local scores when learning the SCEG. One simple reason is that there are comparatively far more scores to calculate for the unrestricted partitions associated with the space of SCEGs than with the restricted set of partitions associated with BNs. However another reason is that when calculating the local scores in a BN for a node $y$ given a set of parents $\text{pa}(y)$, the parent set $\text{pa}(y)$ is an unordered set. This means that on finding the best local score for a node $y$ and a subset of variables $U \subset X \setminus y$, it is also possible to store the best set of parents $\text{pa}(y)^*$ together with the best local score at little extra storage cost. This makes the recovery of the BN after the sink ordering has been found quite straightforward.

In contrast, for SCEGs, whilst it is true that the score of the best scoring partition for a sink variable will be independent of the ordering of the previous variables in the tree, as mentioned earlier, the actual partition will depend on their ordering—different orderings will permute leaves of the nodes at the sink level and hence lead to a permuted partitions. This means that a fast recovery of a SCEG from a variable ordering and a stored list of local scores similar to that as possible for BNs would require storing the partitions for $BLS(i, W)$ for every ordering of $W \subset X \setminus i$. Clearly this is adding a factorial factor of complexity in the calculations required at each level, and we already have a complexity of Bell number order in calculating the local scores for a given ordering. Hence it is computationally simpler though still computationally expensive to apply Algorithm 1 after the optimal ordering has been found.

For problems of a magnitude similar to our running example, the dynamic programming method of this paper seems to be quite adequate for a search of the space to find an optimal. However as the number of atoms in the underlying tree gets larger the method quickly slow down and soon become impractical. If this happens then we would have to exploit options of further constrain the search space using additional domain information if available. Otherwise approximate search methods such as the Bayesian Agglomerate Clustering Algorithm developed for CEGs in (Freeman and Smith, 2011a) would need to be applied, or perhaps hybrid algorithms with optimal clustering of the nodes in the first few levels of a tree $\mathcal{E}^{\prec}$ and greedy clustering when the numbers of partitions in a level exceeds a certain threshold.

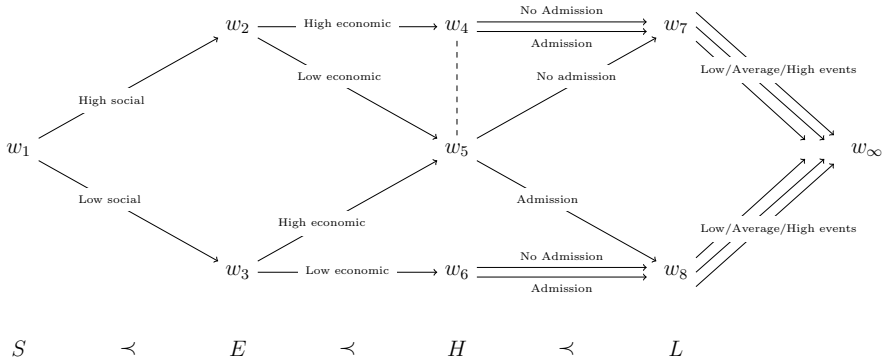


Figure 7: MAP CEG for the CHDS data found with no restriction on the node ordering.

# 6 Christchurch Example

We applied the optimal MAP search algorithms to the CHDS data (a) using Algorithm 1 with the ordering $S \prec E \prec L \prec H$ of the variables employed in the CEG found by (Barclay et al., 2013a), and (b) using Algorithm 3 without assuming an ordering of variables. In both searches we used the same value for the prior parameter $\bar{\alpha} = 3$ as in (Barclay et al., 2013a). When the node ordering was used the optimal SCEG we obtained was the same as that found in (Barclay et al., 2013a) using the greedy search algorithm in (Freeman and Smith, 2011a). It is worth emphasising that this SCEG has a Bayes factor of 79698 in its favour in comparison to the unrestricted MAP optimal BN fitting the data. The SCEG is displayed in Figure 6. When the ordering of variables was not assumed, a MAP SCEG was found with a slightly higher score (by a Bayes factor of 1.139) than the MAP SCEG found under the assumed node ordering. The MAP SCEG ordering is given by $S \prec E \prec H \prec L$, and the corresponding SCEG is shown in Figure 7.

It is interesting to contrast the two differing *causal interpretations* of the SCEGs shown Figure 6 and Figure 7. A detailed discussion of the causal interpretation of Figure 6 was given by (Barclay et al., 2013a), which we summarise here.

Reading Figure 6 from left to right, it appears that the social background $S$ has an effect on the economic situation $E$. The economic situation seems to have no effect on the number of life events $L$ for families from a higher social background. However, in a family from a lower social background the economic situation seems to affect the number of life events that occur. Children from a family of high social background and a low number of life events are in a separate position and hence have a different probability of admissions $H$ to the other individuals. Children from socially advantaged families with an average number of life events are in the same position as children from socially disadvantaged families with a high economic situation and a low or average number of life events, as are children from a low economic situation with a low number of life events. However, all individuals with a high number of life events are in the same position irrespective of their social or economic background, and are in the same position as an individual from a low

social and economic background with only an average number of life events.

In contrast in Figure 7 the ordering of the variables hospital admission $H$ and life events $L$ is reversed. This can also be given a causal interpretation, so that life events come out of circumstances related to hospital admissions, rather than vice versa. For example, families who have to deal with long term illness in children often break up as a consequence. In more detail, a summary of the causal interpretation of Figure 7 can be given as follows. As for Figure 6, it appears that the social background has an effect on the economic situation. The life events for children from socially and economically advantaged children is independent of hospital admissions, and such children are in the same position regarding life events as those from either a high social but low economic position or those from a low social but high economic position *provided* that they do not have a hospital admission. However if they do suffer hospital admission, then these children are in the same position regarding life events as children from low social and economic status regardless of whether or not they have had a hospital admission.

Now recall that the Bayes factor between these two models is only 1.139. Is this sufficient to choose Figure 7 over Figure 6 as a causal explanation of the data? This depends on the *a priori* belief about the causal ordering of the variables, for which there is typically information about whether or not certain orderings (trees) make sense or not. So, for example, if we were to insist that hospital admission $H$ is the final (response) variable (as assumed in the study of Fergusson et al. (1986), then we would select Figure 6 because a priori we give the SCEG of Figure 7 zero probability. Alternatively, if we were to give both orderings the same prior odds, then the SCEG of Figure 7 would emerge as the *MAP* causal explanation.

This example shows that the structural priors assumed can have a strong influence on the causal inference. One could for example assume that *a priori* each SCEG in the search space has the same prior, or that each SCEG in the same equivalence class has the same prior, or that each ordering of the measurement variables has the same prior. Note that if the state spaces of the variables are not all equal, then their ordering will affect the numbers of nodes on the event tree at each level, and hence strongly affect the number of partitions that can be formed from these nodes. For the example in this paper, with the node ordering $S \prec E \prec L \prec H$ the level sizes from the root of the event tree are $(1, 2, 4, 12)$ whilst for the ordering $S \prec E \prec H \prec L$ the level sizes are $(1, 2, 4, 8)$. Hence there are $B_{12}/B_8 \approx 1018$ more SCEGs with the first node ordering than the second. We note when learning BNs we are faced with similar difficult choices for selecting an appropriate structural prior.

The fact that our exploratory technique has led us to discover two different competing causal hypotheses that seem to be well supported might also encourage us to entertain models that lie outside the original class. Thus in our example, an alternative possibility to choosing the MAP SCEG as the unique causal explanation, is to adopt a Bayesian viewpoint that admits that *both* SCEG models provide possible causal explanations of the data, for which one explanation has hospital admissions preceding life events in a small majority of cases, i.e, is the slightly more common explanation. This too appears sensible. For example, in some cases hospital admissions can occur at or soon after birth, before life events have taken place and which thus come later. In other families, life events can take

place much later in the life of a child, leading to hospital admission. This might prompt us no longer to insist on a *single* graphical model to explain the causal dependencies in the data, but instead utilise the asymmetries picked up in the tree to distinguish certain causal orderings and treat them as all viable causal explanations, each pertaining to certain cases.

However one must be very careful in making such a judgement, because we assumed at the outset that our observations are drawn from an homogeneous and not a mixed population which all share the same SCEG. Instead, we might now set up a new search of models which explicitly modelled such heterogeneity. This could be achieved for example by introducing an extra unobserved variable to describe the mixing to allow both possibilities. Of course then the score function would not have the simple decomposable form. Nevertheless, albeit less fast search techniques developed for BNs such as (Spiegelhalter and Cowell, 1992) can be easily adapted to this much smaller and more specific class to discover even better explanations. We note in passing that in (Barclay et al., 2013b) (see the 2 time slice CEG section and subsequent model selection) - dynamic versions of the CEG class can explicitly entertain the two different temporal hypotheses such as the one above and so utilize the time sequence data also available from the study above to better score models within this extended class. The point we make here is that the exploratory analysis we have illustrated above can be used not only used to discover good models within the large searched class, but also to entertain more elaborate explanations of a given process.

# 7 Discussion

One of the attractive features of CEGs, arising as they do from event trees, is that they present a natural framework to discuss causality. In this paper we introduced a restricted class of CEGs, the *stratified chain event graphs* (SCEGs) which we propose as a particularly useful model class for standard types of causal modelling. We presented a dynamical programming method for carrying out an exhaustive search for the highest scoring SCEG under the assumption of complete data and a decomposable score, illustrated using the BDEu score but the method clearly extends to other decomposable scores. We showed that the search algorithm contains the dynamic programming search for optimal scoring BNs as a special case, and also showed how the complexity of search increases dramatically for SCEGs over the complexity of search for BNs in the same number of variables. The dramatic increase in complexity means that the algorithm presented in this paper is limited in practice to a very small number of variables, and that approximate search algorithms such as given by Freeman and Smith (2011a) would be needed for larger problems.

However there may be some scope for extending the exhaustive search possibilities for SCEG by developing methods that have been proposed for learning BNs. For example, it has been shown that certain inequalities for the BDEu score mean that efficient bounds may be placed on the maximum number of parents that a variable can have (de Campos and Ji, 2010). That is, in BN learning it it sometimes possible to show that scores can only decrease if parents are added to a node, hence the optimal network can be found without having to find all possible parent-child scores. Such information, combined with

recent developments in learning BNs using convex optimization techniques (Jaakkola et al., 2010; Cussens, 2011) mean that provably optimal BNs may be found when the number of variables exceeds that which is possible by the exact dynamic programming methods. Speculatively, a SCEG analogue would perhaps be to show the existence of constraints that the optimal partition of a level of an event tree could have, so reducing considerably the space of allowed partitions that the search needs to be carried out on. The development of convex optimization methods for learning CEGs is another promising approach yet be explored.

Another extension that has not been explored in this paper is to consider combining stages at different levels of an event tree. This removes us from the space of SCEGs considered in this paper, which was restricted to only combining stages occurring on the same level. Clearly such an extension will lead to an even higher computational cost than the algorithms presented in this paper, but may lead to uncovering further conditional independence properties within datasets and would seem appropriate for highly asymmetrical problems. However, even when restricted to learning SCEGs, having an initial asymmetrical tree does not pose problems.

The efficacy of selecting across CEGs consistent with a fixed causal ordering has already been demonstrated. However the fuller search methods we described here also enable us to search different putative causal orderings of variables to find higher scoring ones. In the particular example we see how such a search can provoke us to examine new previously unconsidered causal hypotheses and so help in the formulation of better explanations of the observed phenomena. We note that the CEG family is rich enough to distinguish causal orders which under BN models would be impossible to distinguish. Of course to associate these orders as truly "causal" in any formal sense is heroic. In particular it leans even more heavily on the parsimony principle (Pearl, 2009) than BN search does to infer directionality and from this a putative causation. But when model search is used simply to encourage reflection on the nature of the underlying data generating mechanism this search method used on this model class nevertheless provides us with a valuable new exploratory tool. The fact that these full search methods can identify the top scoring models makes the discovery even more compelling.

# Acknowledgements

We would like to thank D. Fergusson for permission to use the data from the Christchurch Health and Development Study.

# References

Andersson, S. A., Madigan, D., and Perlman, M. D. (1996). A characterization of Markov equivalence classes for acyclic digraphs. *Annals of Statistics*, 25:505–541.

Barclay, L. M., Hutton, J. L., and Smith, J. Q. (2012). Chain event graphs for informed missingness. *Bayesian Analysis*, pages 12–17.

Barclay, L. M., Hutton, J. L., and Smith, J. Q. (2013a). Refining a Bayesian Network using cegs. *International Journal of Approximate Reasoning*. In press.

Barclay, L. M., Smith, J. Q., Thwaites, P., and Nicholson, A. (2013b). Dynamic chain event graphs.

Bielza, C. and Shenoy, P. P. (1999). A comparison of graphical techniques for asymmetric decision problems. *Management Science*, 45(11):1552–1569.

Boutilier, C., Friedman, N., Goldszmidt, M., and Koller, D. (1996). Context-specific independence in bayesian networks. In Horvitz, E. and Jensen, F. V., editors, *Proceedings of the 12th Annual Conference on Uncertainty in Artificial Intelligence*, pages 115–123.

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

Cussens, J. (2011). Bayesian network learning with cutting planes. In Cozman, F. G. and Pfeffer, A., editors, *Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence (UAI 2011)*, pages 153–160. AUAI Press.

de Campos, C. and Ji, Q. (2010). Properties of Bayesian Dirichlet scores to learn bayesian network structures. In *Twenty-Fourth AAAI Conference on Artificial Intelligence*.

Fergusson, D., Horwood, L., and Shannon, F. (1986). Social and family factors in childhood hospital admission. *Journal of epidemiology and community health*, 40(1):50–58.

Freeman, G. and Smith, J. Q. (2011a). Bayesian map model selection of chain event graphs. *Journal of Multivariate Analysis*, 102(7):1152–1165.

Freeman, G. and Smith, J. Q. (2011b). Dynamic staged trees for discrete multivariate time series: forecasting, model selection and causal analysis. *Bayesian Analysis*, 6(2):279–305.

French, S. and Insua, D. R. (2000). Statistical decision theory. kendall's library of statistics 9. *Arnold, London*.

Friedman, N. and Goldszmidt, M. (1998). Learning Bayesian networks with local structure. In Jordan, M. I., editor, *Learning in Graphical Models*, pages 421–460. Kluwer Academic Publishers, Dordrecht, The Netherlands.

Geiger, D. and Heckerman, D. (1997). A characterization of the dirichlet distribution through global and local parameter independence. *The Annals of Statistics*, 25(3):1344–1369.

Heckerman, D. (1998). A tutorial on learning with Bayesian networks. In Jordan, M. I., editor, *Learning in Graphical Models*, pages 301–354. Kluwer Academic Publishers, Dordrecht, The Netherlands.

Hugin Expert A/S (2012). *Hugin API Reference Manual, Version 7.6*. Hugin Expert A/S, Aalborg, Denmark.

Jaakkola, T., Sontag, D., Globerson, A., and Meila, M. (2010). Learning Bayesian network structure using lp relaxations. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics: May 13-15, 2010, Sardinia, Italy*, pages 358–365. Society for Artificial Intelligence and Statistics.

Jaeger, M. (2004). Probabilistic decision graphs—combining verification and AI techniques for probabilistic inference. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 12(supp01):19–42.

Koivisto, M. and Sood, K. (2004). Exact Bayesian structure discovery in Bayesian networks. *Journal of Machine Learning Research*, 5:549–573.

Koller, D. and Pfeffer, A. (1997). Object-oriented Bayesian networks. In Geiger, D. and Shenoy, P., editors, *Proceedings of the 13th Annual Conference on Uncertainty in Artificial Intelligence*, pages 302–313, San Francisco, California. Morgan Kaufmann.

Maathuis, M. H., Colombo, D., Kalisch, M., and Bühlmann, P. (2010). Predicting causal effects in large-scale systems from observational data. *Nature Methods*, 7(4):247–248.

McAllester, D., Collins, M., and Pereira, F. (2004). Case-factor diagrams for structured probabilistic modeling. In *Proceedings of the 20th conference on Uncertainty in Artificial Intelligence*, UAI '04, pages 382–391, Arlington, Virginia, United States. AUAI Press.

Oliver, J. J. (1993). Decision graphs—an extension of decision trees. In *Proceedings of the Fourth International Workshop on Artificial Intelligence and Statistics: January 3-6, 1993, Ft. Lauderdale, Florida*, pages 343–350. Extended version available as TR-173, Department of Computer Science, Monash University, Australia.

Olmsted, S. M. (1983). *On Representing and Solving Decision Problems*. Ph.D. Thesis, Department of Engineering–Economic Systems, Stanford University, Stanford, California.

Pearl, J. (2009). Cambridge University Press.

Peer, D., Regev, A., Elidan, G., and Friedman, N. (2001). Inferring subnetworks from perturbed expression profiles. *Bioinformatics*, 17(suppl 1):S215–S224.

Ramsey, J. D., Hanson, S. J., and Glymour, C. (2011). Multi-subject search correctly identifies causal connections and most causal directions in the dcm models of the smith et al. simulation study. *NeuroImage*, 58(3):838–848.

Sachs, K., Perez, O., Pe'er, D., Lauffenburger, D. A., and Nolan, G. P. (2005). Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308(5721):523–529.

Salmerón, A., Cano, A., and Moral, S. (2000). Importance sampling in bayesian networks using probability trees. *Computational Statistics & Data Analysis*, 34(4):387–413.

Shafer, G. (1996). *The art of causal conjecture*. MIT Press.

Silander, T. and Myllymäkki, P. (2006). A simple approach to finding the globally optimal Bayesian network structure. In Dechter, R. and Richardson, T., editors, *Proceedings of the* $22^{nd}$ *Conference on Artificial Intelligence (UAI 2006)*, pages 445–452. AUAI Press.

Singh, A. P. and Moore, A. W. (2005). Finding optimal Bayesian networks by dynamic programming. Technical Report CMU-CALD-05-106, Carnegie Mellon University.

Smith, J. Q. (2010). *Bayesian decision analysis: principles and practice*. Cambridge University Press.

Smith, J. Q. and Anderson, P. E. (2008). Conditional independence and chain event graphs. *Artificial Intelligence*, 172(1):42–68.

Smith, J. Q. and Freeman, G. (2011). Distributional kalman filters for bayesian forecasting and closed form recurrences. *Journal of Forecasting*, 30(1):210–224.

Spiegelhalter, D. J. and Cowell, R. G. (1992). Learning in probabilistic expert systems. In Bernardo, J. M., Berger, J. O., Dawid, A. P., and Smith, A. F. M., editors, *Bayesian Statistics 4*, pages 447–465, Oxford, United Kingdom. Clarendon Press.

Spirtes, P., Glymour, C., and Scheines, R. (1993). *Causation, Prediction, and Search*. Springer–Verlag, New York.

Thwaites, P. (2013). Causal identifiability via chain event graphs. *Artificial Intelligence*, pages 291–315.

Thwaites, P., Freeman, G., and Smith, J. Q. (2009). Chain event graph map model selection. In *Proceedings of KEOD 09 Madeira*, pages 392–395.

Thwaites, P. and Smith, J. (2011). Separation theorems for chain event graphs.

Thwaites, P., Smith, J. Q., and Cowell, R. G. (2008). Propagation using chain event graphs. In McAllester, D. and Myllymäki, P., editors, *Proceedings of the 24th Conference in Uncertainty in Artificial Intelligence, Helsinki, July 2008*, pages 546–553.

Thwaites, P., Smith, J. Q., and Riccomagno, E. (2010). Causal analysis with chain event graphs. *Artificial Intelligence*, 174(12):889–909.

Verma, T. and Pearl, J. (1991). Equivalence and synthesis of causal models. In Bonissone, P. P., Henrion, M., Kanal, L. N., and Lemmer, J. F., editors, *Uncertainty in Artificial Intelligence 6*, pages 255–268. North-Holland, Amsterdam, The Netherlands.

Wallace, C. S. and Patrick, J. (1993). Coding decision trees. *Machine Learning*, 11(1):7–22.