Beyond Conjugacy for Chain Event Graph Model Selection

Aditi Shenvi
Department of Statistics, University of Warwick
and
Silvia Liverani
School of Mathematical Sciences, Queen Mary University of London

July 12, 2024

Abstract

Chain event graphs are a family of probabilistic graphical models that generalise Bayesian networks and have been successfully applied to a wide range of domains. Unlike Bayesian networks, these models can encode context-specific conditional independencies as well as asymmetric developments within the evolution of a process. More recently, new model classes belonging to the chain event graph family have been developed for modelling time-to-event data to study the temporal dynamics of a process. However, existing Bayesian model selection algorithms for chain event graphs and its variants rely on all parameters having conjugate priors. This is unrealistic for many real-world applications. In this paper, we propose a mixture modelling approach to model selection in chain event graphs that does not rely on conjugacy. Moreover, we show that this methodology is more amenable to being robustly scaled than the existing model selection algorithms used for this family. We demonstrate our techniques on simulated datasets.

Keywords: Mixture models, staged trees, graphical models, time-to-event analysis

1 Introduction

Chain event graphs (CEGs) are a family of probabilistic graphical models that were first proposed in Smith and Anderson (2008) as an alternative to Bayesian networks (BNs). In particular, CEGs were developed to explicitly accommodate processes exhibiting asymmetries of two types: (1) asymmetric independence structures or *context-specific* conditional independences where some statistical independences hold for certain values of the conditioning variables but not the others; and (2) asymmetric event spaces which are precisely event spaces that do not admit a product space structure. The latter asymmetry arises due to the presence of structural zeros and structural missing values, often-times by design (Shenvi and Smith, 2020). For example, consider modelling hospitalisations arising from infection caused by a circulating virus, and suppose that one of the two strains (call it strain A) of the virus has no treatment currently available while the other has a choice of two possible treatments. On the one hand, a variable of "Treatment" with state space $\{\text{Treatment 1}, \text{Treatment 2}\}$ would be structurally missing and have no sensible value for those infected by strain A of the virus. Whereas on the other hand, if its state space is redefined to be $\{\text{Treatment 1}, \text{Treatment 2}, \text{No treatment}\}$ then Treatment 1 and Treatment 2 would have structurally zero counts for those infected by strain A, i.e. irrespective of the sample size, there would always be zero individuals who are treated with either Treatment 1 or Treatment 2 among those infected by strain A. Such a process is inherently asymmetric. BNs, being variable-based – i.e. they use variables as the building blocks of their models – are unable to fully describe such asymmetries within their underlying statistical model and graphical structure. The CEG for a process, on the other hand, is obtained through a transformation of an event tree describing the process and thus, has an event-based[1] topology. This event-based formulation enables a CEG to fully embed structural asymmetries within its model and graph. In fact, in order to accommodate context-specific conditional

[1] An event is an element or a subset of elements of the state space of a variable.

independences within a BN model, the modifications proposed in the literature typically rely on tree-based structures (e.g. Boutilier et al. (1996); Poole and Zhang (2003); Jabbari et al. (2018)), and further, by design, BNs are unable to explicitly encode asymmetric event spaces.

The parameters of a *non-temporal CEG*[2] are given by the parameters of the conditional transition distributions for the nodes in its graph. These distributions govern which event occurs next given that a particular node has been reached. More recently, new classes of the temporal CEGs have been proposed for modelling time-to-event data to study the temporal dynamics of a process (Shenvi, 2021; Shenvi and Smith, 2019; Barclay et al., 2015; Collazo and Smith, 2018). Temporal CEGs introduce conditional holding time random variables for the transitions modelled by the process. These random variables describe how long it takes for the next event to occur given that a particular node has been reached. For instance, if we consider the infection example introduced earlier, a temporal CEG would be suitable if we are not only interested in studying the evolution of the infection-to-hospitalisation trajectory of individuals but also how long it takes for the various transitions to occur in each trajectory. Thus, a model belonging to one of these classes has precisely two categories of parameters; one for modelling the conditional transition distributions, and the other for modelling the conditional holding time distributions. Note that the model selection exercise in these CEGs can be done independently for each category of random variable under the standard assumption of parameter independence.

The existing Bayesian model selection algorithms employed for non-temporal and temporal CEGs (Silander and Leong, 2013; Cowell and Smith, 2014; Freeman and Smith, 2011; Shenvi, 2021; Strong and Smith, 2022a) rely on the parameters of the conditional transition and conditional holding time distributions having conjugate priors. For instance, the Binomial or Multinomial distributions are typically used for the conditional transition dis-

[2] In the CEG literature, a 'CEG' often refers to the simplest discrete state space class of this family. Here, we refer to this as a 'non-temporal CEG' to distinguish it from other classes of the CEG family.

tributions whereas the Weibull distribution with known shape parameter is used for the conditional holding times.

Whilst conjugacy of prior and posterior distributions of parameters is desirable for its closed form analytical solutions and for the interpretability it lends to the hyperparameters, conjugate settings are either infeasible or inappropriate in most cases. Under the setting of sampling with replacement from a given population size with a fixed and finite number of categories, the Multinomial distribution (or equivalently, the Binomial distribution when the number of categories is 2) is perhaps the most appropriate choice for the conditional transition distributions (Minka, 2003). However, there is no reason why the conditional holding time distributions need to belong to the conjugate family. A simple example here is that even if we believe the conditional holding times to be governed by a Weibull distribution, it is typically unlikely that we know the shape parameter of this distribution. Thus, the conjugacy requirement is less restrictive for non-temporal CEGs than for temporal CEGs.

Moreover, even without consideration of the conjugacy issue, the existing model selection approaches are not easily scalable or are not robust when scaled. The two main existing approaches to Bayesian model selection in non-temporal and temporal CEGs come from using a Bayesian scoring rule in (1) a dynamic programming approach combined with brute-force partition scoring for a globally optimal model and (2) the agglomerative hierarchical clustering (AHC). The brute-force element of the former approach is clearly not scalable, whereas the AHC is a greedy algorithm that can be scaled relatively well but is not robust. Note that non-Bayesian alternatives for model selection have been developed, including non-Bayesian scoring rules with the two approaches above (see e.g. Silander and Leong (2013); Carli et al. (2022)).

In this paper, we propose a novel methodology for model selection in CEGs which casts the model selection problem into the problem of fitting a mixture model. We demonstrate that this simple change of perspective on the problem allows us to use well-developed and well-tested existing software such as Stan (Carpenter et al., 2017) to support model selection in temporal CEGs with non-conjugate conditional holding time distributions. Further, we demonstrate how this approach enables a more robust scaling of model selection for non-temporal and temporal CEGs, compared to the existing model selection algorithms, for conditional transition distributions when these follow the Binomial distribution (also known as *binary trees*). Strong and Smith (2022b) describes how any event tree can be recast as a binary tree and thus, this is not a major restriction. Thus, our paper vastly extends the range of applications that can be supported by the CEG family and also opens new avenues to extend their applicability.

This paper is organised as follows. In Section 2 we review non-temporal and temporal CEGs, and the model selection algorithms employed for these within the literature. In Section 3 we describe how the model selection problem can be posed as a mixture modelling problem and discuss its advantages. In Section 4 we illustrate this methodology through simulated examples. We conclude with a discussion in Section 5.

2 Preliminaries

2.1 Chain Event Graphs

CEGs are an event-based probabilistic graphical modelling family that describe the evolution of a process through a sequential unfolding of events. They harness the symmetries within the process to provide a compact representation of the process. Crucially, through their event-based formulation, they are able to embed asymmetric independence structures and asymmetric event spaces within their statistical models and graphs; see Collazo et al. (2018); Shenvi et al. (2018); Shenvi (2021).

The construction of a CEG model begins by eliciting an event tree description of the process from a combination of domain experts, existing literature and data. Event trees provide a natural framework for describing the step-by-step evolution of a process – an excellent exposition of trees and their fundamental role in probability theory and causality can be found in Shafer (1996). A non-technical summary (Shenvi and Smith, 2020) of the transitions an event tree must go through to become the graph of a CEG model are given below:

- Nodes in the event tree whose one-step-ahead evolutions are equivalent – in terms of the conditional transition distributions for non-temporal CEGs and the conditional transition and conditional holding time distributions for the temporal CEGs – are said to be in the same *stage* and are assigned the same colour to indicate their shared stage membership.
- Nodes whose rooted subtrees (i.e. the subtree obtained by considering that node as the root) are isomorphic, in the structure and colour preserving sense, are said to be in the same *position* and are merged into a single node which retains the colouring of its merged nodes.
- All the leaves of the tree are merged into a single node called the *sink* node.

The simplest CEG class (Collazo et al., 2018), which we refer to as the non-temporal CEG here, explicitly models the conditional transition distributions but not the conditional holding time distributions – these are included implicitly through its Markov assumption (Shenvi, 2021). Newer CEG classes such as the dynamic CEG (Barclay et al., 2015; Collazo and Smith, 2018), extended dynamic CEG (Barclay et al., 2015) and the continuous-time dynamic CEG (Shenvi and Smith, 2019; Shenvi, 2021) were proposed for modelling longitudinal temporal processes with asymmetries. These classes explicitly model the conditional holding time distributions. We note here that these model classes can also be defined over non-longitudinal temporal processes – which we define here to be a temporal process whose underlying event tree description is finite in its number of nodes and edges. For simplicity of illustration, in this paper we focus on these non-longitudinal temporal CEGs, referred to simply as temporal CEGs. The model selection approach described in this paper extends to dynamic temporal CEGs in a straightforward way. Further, under the standard assumption of parameter independence described in Section 2.2, the model selection exercise simplifies into two independent clustering problems; one for the conditional transition distributions and the other for the conditional holding time distributions. Therefore, we will focus on temporal CEGs with the understanding that our model selection approach for conditional transition distributions can be applied directly to non-temporal CEGs as well.

Denote by $\mathcal{T}$ an event tree with a finite node set $V(\mathcal{T})$ and a directed edge set $E(\mathcal{T})$. Each edge $e \in E(\mathcal{T})$ is an ordered triple of the type (v, v', l) denoting that e emanates from node v, terminates in node v' and has edge label l. The set of leaves in $\mathcal{T}$ is denoted by $L(\mathcal{T})$, and the non-leaf nodes known as *situations* are represented by the set $S(\mathcal{T}) = V(\mathcal{T}) \backslash L(\mathcal{T})$. The set of children of a node v is denoted by $\text{ch}(v)$. Let $\boldsymbol{\Phi}_{\mathcal{T}} = \{\boldsymbol{\theta}_v | v \in S(\mathcal{T})\}$ where $\boldsymbol{\theta_v} = (\theta(e) | e = (v, v', l) \in E(\mathcal{T}), v' \in \text{ch(v)})$ denotes the conditional transition parameters for each node $v \in S(\mathcal{T})$.

Each transition from node v to v' along some edge $e = (v, v', l)$ between them is associated with a holding time which indicates the time spent in node v before transitioning along e to v'. Denote this conditional holding time by variable $H(e)$. Here we assume that the holding time is dependent on both the current situation and the situation visited next. However, the conditional transition probabilities are independent of the holding times. Let $\boldsymbol{\mathcal{H}}_{\mathcal{T}} = \{\mathbf{H}(v) | v \in S(\mathcal{T})\}$ where $\mathbf{H}(v) = \{H(e) | e = (v, v', l) \in E(\mathcal{T}), v' \in \text{ch(v)}\}$ denotes the set of holding time variables for each edge emanating from situation $v \in S(\mathcal{T})$. Note that we assume here that all transitions in the event tree are associated with a holding time. Some temporal processes – in particular, those including time-invariant covariates in their description – might have some transitions for which a holding time is illogical. See Shenvi (2021), pp 87 – 89, for a description of how these can be accommodated.

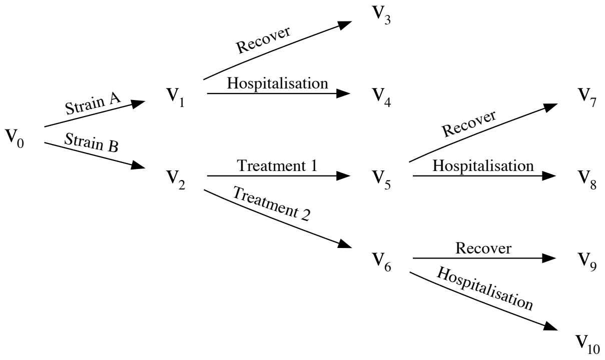


Figure 1: Event tree for the infection process in Example 1.

Example 1 (Infection example) *Consider the infection example described earlier. Suppose we are studying hospitalisations occurring due to infection from one of two strains (strains A and B) of a circulating virus. Suppose that research showed that the available treatments are effective against an infection caused by strain B of the virus but not strain A. Therefore, individuals infected with strain A have no treatment options available whereas those infected with strain B have the options of treatment 1 and treatment 2. Thus, the treatment variable is structurally missing for individuals infected by strain A. The outcome of interest for this process is either recovery or hospitalisation. This process is structurally asymmetric and can be described by the event tree in Figure 1. Here, for situation* $v_2 \in S(\mathcal{T})$ *we have emanating edges* $(v_2, v_5,$ *Treatment 1*$)$ *and* $(v_2, v_6,$ *Treatment 2*$)$*, and its children are nodes* v_5 *and* v_6*. The random variables* $H(v_2, v_5,$ *Treatment 1*$)$ *and* $H(v_2, v_6,$ *Treatment 2*$)$ *describe the duration of the treatment after being infected by strain B for treatments 1 and 2 respectively.*

Definition 2 (Stage) *In an event tree* $\mathcal{T}$*, two situations* v *and* v' *are said to be in the same stage whenever*

- $\boldsymbol{\theta}_v = \boldsymbol{\theta}_{v'}$ *such that, for edges* e *and* e' *emanating from* v *and* v' *respectively with* $\theta(e) = \theta(e')$*, we require that* $e = (v, \cdot, l)$ *and* $e' = (v', \cdot, l)$ *for some edge label* l *and* *where $\cdot$ is a placeholder for any second vertex,*

- *Variables $H(e)$ and $H(e')$ for $e = (v, \cdot, l)$ and $e' = (v', \cdot, l)$ follow the same distribution.*

Definition 3 (Hyperstage) *A hyperstage for an event tree $\mathcal{T}$ is a collection of sets such that two situations cannot be in the same stage unless they belong to the same set in the hyperstage.*

Throughout this paper, our examples use hyperstages with mutually exclusive sets where each set contains the situations belonging to a specific variable. Situations belonging to the same stage are given the same colour to represent the shared membership. An event tree $\mathcal{T}$ whose situations are coloured according to their stage memberships is called a *staged tree* and is denoted as $\mathcal{S}$. The collection of stages $\mathbb{U}$ partitions the set of situations $S(\mathcal{T})$. It is common practice to colour trivial, i.e. singleton, stages black to prevent visual cluttering.

Situations in the staged tree whose rooted subtrees are isomorphic have equivalent sets of edge labels, conditional transition parameters, and conditional holding time distributions[3]. Situations whose rooted subtrees are isomorphic are said to belong to the same *position*. Denote the collection of positions by $\mathbb{W}$. Observe that $\mathbb{W}$ creates a finer partition of $S(\mathcal{T})$. We can now define a temporal CEG as follows.

Definition 4 (Temporal Chain Event Graph) *A temporal CEG $\mathcal{C} = (V(\mathcal{C}), E(\mathcal{C}))$ is defined by the tuple $(\mathcal{S}, \mathbb{W}, \boldsymbol{\Phi}_{\mathcal{S}}, \boldsymbol{\mathcal{H}}_{\mathcal{S}})$ with the following properties:*

- *$V(\mathcal{C}) = R(\mathbb{W}) \cup w_\infty$ where $R(\mathbb{W})$ is the set of situations representing each position set in $\mathbb{W}$ and w_∞ is the sink node. Additionally, nodes in $R(\mathbb{W})$ retain their stage colouring and for $w \in R(\mathbb{W})$, $\theta_{\mathcal{C}}(w) = \theta_{\mathcal{S}}(w)$ and $\boldsymbol{H}_{\mathcal{C}}(w) = \boldsymbol{H}_{\mathcal{S}}(w)$.*
- *Situations in $\mathcal{S}$ belonging to the same position set in $\mathbb{W}$ are contracted into their representative node contained in $R(\mathbb{W})$. This node contraction merges multiple edges between two nodes into a single edge only if they share the same edge label.*

[3] In a non-technical sense, this implies that v and v' have identical future evolutions.

- *Leaves of $\mathcal{S}$ are contracted into sink node w_∞.*

Example 5 (Infection example (continued)) *Suppose that the probability of recovery is independent of the treatment, given infection by strain B. This is a form of context-specific information which can be expressed as*

$$\mathit{Outcome} \perp\!\!\!\perp \mathit{Treatment} \,|\, \mathit{Strain} = \mathit{Strain\ B}$$

where $\perp\!\!\!\perp$ stands for probabilistic independence and the vertical bar shows conditioning variables on the right. Suppose also that $H(v_1, v_3, \mathit{Recover})$, $H(v_5, v_7, \mathit{Recover})$ and $H(v_6, v_9, \mathit{Recover})$ follow the same distribution, as do $H(v_1, v_4, \mathit{Hospitalisation})$, $H(v_5, v_8, \mathit{Hospitalisation})$ and $H(v_6, v_{10}, \mathit{Hospitalisation})$. The stage partition here is given by $\mathbb{U}$ which contains the following sets:

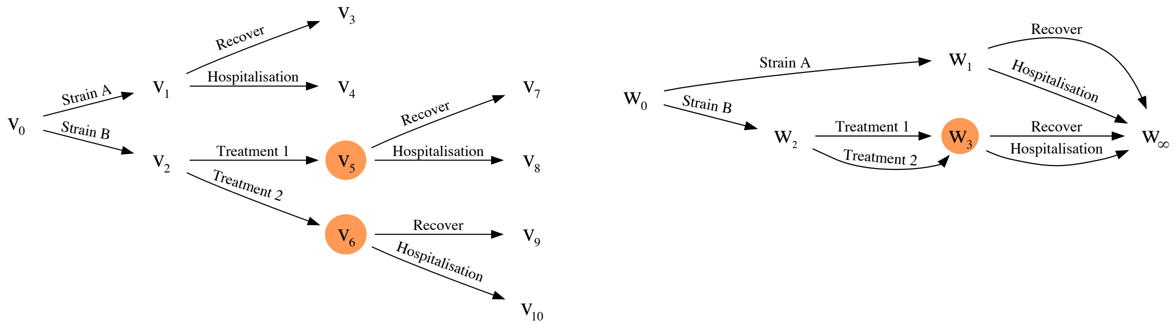
$$\{v_0\}, \{v_1\}, \{v_2\}, \{v_5, v_6\}.$$

Observe that v_1 is not in the same stage as v_5 and v_6 as although it satisfies the second condition given in Definition 2, it does not satisfy the first. Figure 2(a) gives the staged tree for this process. In this example, the position partition $\mathbb{W}$ is equivalent to the stage partition $\mathbb{U}$. The leaves v_3, v_4, v_7, v_8, v_9 and v_{10} are combined into a single sink node in the CEG as shown in Figure 2(b).

2.2 Separation of Likelihood

We now demonstrate the conditions under which the parameters of the conditional transition and conditional holding time distributions can be learned independently. This separation of likelihood was first presented in Barclay et al. (2015).

Consider a temporal CEG $\mathcal{C}$ with collection of stages $\mathbb{U} = \{u_1, u_2, \ldots, u_k\}$. Suppose that each stage u_i has k_i emanating edges (i.e. $|\mathrm{ch}(v_i)| = k_i$ for $v_i \in u_i$). Suppose we have a



(a) Staged tree

(b) CEG

Figure 2: Staged tree and CEG for the infection process in Example 5.

complete (ie. without missing values) random sample of n individuals. For each individual $1 \leq m \leq n$, let their data be given by the following sequence of tuples:

$$\rho_m = ((e_{j_1k_1}, h_{j_1k_1}), (e_{j_2k_2}, h_{j_2k_2}), \ldots (e_{j_{l_m}k_{l_m}}, h_{j_{l_m}k_{l_m}})),$$

where the first element of each tuple represents the edge traversed by the individual and the second element gives the holding time associated with that edge.

Denote the summary of the data associated with each stage u_i in the sample by $\mathbf{d}_i = (\mathbf{d}_{i1}, \mathbf{d}_{i2}, \ldots, \mathbf{d}_{ik_i})$ and $\mathbf{h}_i = (\mathbf{h}_{i1}, \mathbf{h}_{i2}, \ldots, \mathbf{h}_{ik_i})$. Here, each $\mathbf{d}_{ij}$ is a vector of ones of length $|\mathbf{d}_{ij}|$ where $|\mathbf{d}_{ij}|$ is the total number of individuals in the sample who traverse the jth edge of stage u_i. Correspondingly, $\mathbf{h}_{ij}$ is a vector of the holding times for the jth edge of stage u_i for each of the $|\mathbf{d}_{ij}|$ individuals in the sample who traverse this edge.

The data from the n individuals can now be summarised for the CEG as $\mathbf{y} = \{\mathbf{y}_1,\ \mathbf{y}_2, \ldots, \mathbf{y}_k\}$ where $\mathbf{y}_i = (\mathbf{d}_i, \mathbf{h}_i)$ corresponds to the data for stage set u_i, $i = 1, 2, \ldots, k$.

Let the conditional transition parameters for stage u_i be given by $\boldsymbol{\theta}_i = \{\theta_{i1}, \theta_{i2}, \ldots, \theta_{ik_i}\}$ and let $\boldsymbol{\Phi}_{\mathcal{C}} = \{\boldsymbol{\theta}_i | u_i \in \mathbb{U}\}$. Let the conditional holding time random variable for the jth edge emanating from stage u_i be parametrised by π_{ij}. Then $\boldsymbol{\pi}_i = \{\pi_{i1}, \pi_{i2}, \ldots, \pi_{ik_i}\}$ is the vector of holding time parameters for stage u_i. Let $\boldsymbol{\Pi}_{\mathcal{C}} = \{\boldsymbol{\pi}_i | u_i \in \mathbb{U}\}$. The likelihood of the temporal CEG $\mathcal{C}$ can be decomposed into a product of the likelihood of each stage as follows:

$$p(\mathbf{y}|\mathbf{\Phi}_{\mathcal{C}}, \mathbf{\Pi}_{\mathcal{C}}, \mathcal{C}) = \prod_{i=1}^{k} p(\mathbf{y}_i|\boldsymbol{\theta}_i, \boldsymbol{\pi}_i, \mathcal{C}). \tag{1}$$

We assume here that the conditional transition and conditional holding time parameters are *a priori* mutually independent. It follows under the separability of the likelihood above that they will also be independent *a posteriori*. With this we can write

$$\begin{aligned} p(\mathbf{y}_i|\boldsymbol{\theta}_i, \boldsymbol{\pi}_i, \mathcal{C}) &= \prod_{j=1}^{k_i} p(\mathbf{d}_{ij}, \mathbf{h}_{ij}|\theta_{ij}, \pi_{ij}, \mathcal{C}) \\ &= \prod_{j=1}^{k_i} p(\mathbf{h}_{ij}|\pi_{ij}, \mathcal{C}) p(\mathbf{d}_{ij}|\theta_{ij}, \mathcal{C}) \\ &= \prod_{j=1}^{k_i} \prod_{l=1}^{|\mathbf{d}_{ij}|} \Big\{ p(h_{ijl}|\pi_{ij}, \mathcal{C}) \times p(d_{ijl}|\theta_{ij}, \mathcal{C}) \Big\}. \end{aligned} \tag{2}$$

Thus the likelihood of the model separates into the likelihoods of the conditional transition and conditional holding time parameters. This conveniently allows us to estimate the conditional transition and conditional holding time parameters independently. This holds irrespective of whether the conditional holding time variables are discrete or continuous. In the simulations in Section 4, we demonstrate our methods for continuous conditional holding time variables.

2.3 CEG Model Selection

Model selection algorithms for temporal CEGs take as input the event tree $\mathcal{T}$ of the process and output the staged tree $\mathcal{S}$ for the process. A temporal CEG $\mathcal{C}$ is uniquely and completely specified by its staged tree and the parameters over the staged tree $\mathbf{\Phi}_{\mathcal{S}}$ and $\boldsymbol{\mathcal{H}}_{\mathcal{S}}$ (Shenvi and Smith, 2020). Hence, the process of model selection in temporal CEGs is equivalent to identifying the collection of stages in its underlying event tree, which itself is identical to clustering the nodes of the event tree. Further, from Section 2.2, we can see that the process of clustering the nodes of the event tree can be split into two parts:

1. **Identifying the situation clusters:** This refers to the first condition of a stage

in Definition 2. Here we aim to identify which sets of situations have equivalent conditional transition parameters.

2. **Identifying the edge clusters:** This refers to the second condition of a stage in Definition 2. Here we aim to identify which sets of edges follow the same conditional holding time distribution.

Additionally, when the sets in the hyperstage are mutually exclusive, the situation and edge clustering can be performed independently over each set; see Shenvi et al. (2018) for an illustration.

The Bayesian CEG model selection algorithm proposed in the literature is the score-based greedy agglomerative hierarchical clustering (AHC) (Freeman and Smith, 2011; Shenvi and Smith, 2019). Under this approach, the aim is to maximise a chosen score function. In the literature, this has generally taken the form of the log marginal likelihood score. The log marginal likelihood can be obtained analytically within the setting of conjugate priors for the conditional transition and conditional holding time distributions. Below we briefly outline the main steps involved in the AHC algorithm. This approach can be similarly applied to identifying the edge clusters.

The AHC algorithm is a local greedy search algorithm which aims to maximise the overall score by finding the next move that leads to a maximum increase in the score. It uses a bottom-up hierarchical clustering methodology beginning with the finest clustering treating each situation as a singleton cluster and successively merging pairs of clusters until the log marginal likelihood score cannot be improved further. The advantage of this approach is that it is fast when the number of situations is small or moderate, for example taking around 100 seconds for 200 situations. We included some timings in Section 4. However, it is difficult to scale due to its cubic time complexity (Nielsen, 2016). Moreover, it does not scale robustly as it only searches a limited area of the model search space and can get stuck in a local maxima. For instance, a temporal CEG for a certain ordering of

4 binary variables – each with the same set of edge labels – has approximately 1.38×10^9 possible stagings but the AHC evaluates only 560 of them at most. In particular, once the AHC merges two situations into the same stage, it cannot undo this. Therefore, as the AHC algorithm is scaled, it tends to produce a large number of spurious clusters as we demonstrate later in Section 4.1. Leonelli and Varando (2022) showed that by constraining the search space and using a BN to initialise the staging, the AHC for event trees can be scaled robustly. However, this approach has not yet been extended to asymmetric event trees.

The AHC approach is designed for event trees that have a fixed variable ordering. In contrast, another approach for model selection is dynamic programming designed for event trees without a fixed variable ordering. Here, learning the variable ordering is part of the model selection process. As described in Cowell and Smith (2014), the decomposability of the marginal likelihood score of the CEG enables us to decompose the larger problem of identifying the variable ordering and the situation clusters into a iterative smaller problems that proceeds as follows:

- The best-scoring clustering for each variable is identified via a brute-force approach by assuming that it is the first variable in the ordering (i.e. starting from the root).
- The variable with the highest score is chosen to be the first variable.
- This process is then repeated for the penultimate variable until we reach the sink.

The brute-force component of the dynamic programming approach is computationally very expensive. To see this, observe that the number of partitions to be evaluated for a layer with k situations is given by the kth Bell number (Cowell and Smith, 2014) which grows exponentially fast in k.

In summary, the AHC is fast and Bayesian but has poor performance and lacks robust scalability, whereas the dynamic programming approach has good performance but is slow. The latter approach is infeasible for all but the smallest of event trees and hence, we do not consider it further in this paper as our focus is on scalability. Our mixture modelling approach, being applicable to event trees with fixed variable orderings, is directly comparable to the AHC. It offers a balance between scalability, speed and performance, and has the added benefit of not needing conjugacy.

3 Mixture Models for CEG Model Selection

In this section, we propose our novel model selection approach, based on mixture models, for temporal CEGs. This approach overcomes the limitation of assuming conjugate settings for the conditional transition and conditional holding time distributions, and is more amenable to robust scaling than the AHC algorithm described in Section 2.3.

3.1 Mixture Models

We first briefly describe a finite mixture model. For an excellent exposition of finite mixture models see Frühwirth-Schnatter (2006). Consider a population with K subgroups where each subgroup k is of relative proportion ℓ_k, for $k = 1, 2, \dots, K$. Hence, $\sum_{k=1}^{K} \ell_k = 1$. Let $\boldsymbol{\ell} = \{\ell_1, \ell_2, \dots, \ell_K\}$. Suppose that the interest lies in modeling a random feature Y such that Y is heterogeneous across the subgroups but homogeneous within each subgroup. Hence, each subgroup k can be associated with a parameter φ_k for the distribution modeling Y; i.e. the distribution of Y for subgroup k is given by $p(Y = y \,|\, \varphi_k)$. Let $\boldsymbol{\varphi} = \{\varphi_1, \varphi_2, \dots, \varphi_K\}$.

Denote by $\mathbf{y} = \{y_1, y_2, \dots, y_n\}$ a random sample of feature Y recorded from this population. Let an indicator variable $\boldsymbol{z}_i = (z_i^1, z_i^2, \dots, z_i^k)$ denote the subgroup occupied by an individual i who is associated with the observation y_i. This gives us

$$z_i^k = \begin{cases} 1, & \text{if } y_i \text{ comes from mixture component } k, \\ 0, & \text{otherwise.} \end{cases}$$

Assuming random sampling from the population, the probability that an individual belongs to subgroup k, for $1 \leq k \leq K$ is given by the Categorical distribution $Cat(\boldsymbol{\ell})$.

Typically, when we sample randomly from this population, we may not know which subgroup the individual belongs to. This could happen because of several reasons such as due to the way the data was collected or due to the subgroups being latent characteristics. The marginal density of $\mathbf{y}$ here is given by the following mixture density

$$p(\mathbf{y}) = \prod_{i=1}^{n} \sum_{k=1}^{K} \ell_k p(y_i \,|\, \varphi_k). \tag{3}$$

which must be numerically approximated (Frühwirth-Schnatter, 2006).

We can evaluate the posterior probability of observation y_i, for an individual i, belonging to subgroup k as follows

$$p(z_i^k = 1|y_i) = \frac{\ell_k p(y_i \,|\, \varphi_k)}{\sum_{j=1}^{K} \ell_j p(y_i \,|\, \varphi_j)}. \tag{4}$$

The above equation results in a soft clustering of the individuals. However, for most applications using CEGs, we are interested in a hard clustering. There are several ways of arriving at a hard clustering. In this paper, for posterior allocation of each individual i to a single subgroup, we can choose the allocation as

$$z_i^* = \underset{k \in \{1,2,\ldots,K\}}{\arg\max}\; p(z_i^k = 1 \,|\, y_i). \tag{5}$$

3.2 CEG Model Selection Approach Based on Mixture Models

We now describe how model selection in CEGs can be cast as a mixture model.

3.2.1 Identifying the Situation Clusters

Consider an event tree $\mathcal{T}$ with n situations each with m outgoing edges and the same set of edge labels. For situation $v_i \in S(\mathcal{T})$, let its associated data vector be given by $\mathbf{y}_i = (y_{i1}, y_{i2}, \ldots, y_{im})$ where y_{ij} represents the number of individuals in the random sample that arrive at situation v_i and traverse its jth emanating edge, for $1 \leq i \leq n$ and $1 \leq j \leq m$.

Here $\mathbf{y} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n\}$ is the data vector and $\boldsymbol{\theta} = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_n\}$ is the parameter vector where $\boldsymbol{\theta}_i$ represents the conditional transition parameter vector for situation v_i.

The model selection problem can be described as identifying the number and composition of the situation clusters in $\mathcal{T}$. For a fixed number of situation clusters, this simplifies to fitting a standard finite mixture model as described by Equation 3. However, generally the number of situation clusters within a given event tree is unknown. To overcome this problem, we propose here an approach motivated by the AHC algorithm described in Section 2.3. However, instead of a bottom-up approach like the AHC, we take a top-down approach[4] as this generally results in a relatively conservative number of clusters. We start with fitting a mixture model with two clusters/components and then sequentially increase the number of components as long as there is an improvement in the log marginal likelihood score of the model, which naturally penalizes models with more components (and more parameters) (Berger and Jefferys, 1992). Recall that log marginal likelihood of a finite mixture model with two or more components is not available analytically. Instead, we estimate it using bridge sampling (Gronau et al., 2017). A simplified pseudo-code of the proposed model selection algorithm is presented in Algorithm 1.

Whilst the above algorithm can easily handle several hundreds of situations for a fixed number of components, it will be significantly slowed down by fitting the mixture model for several potential numbers of components. As with the dynamic programming approach, the run time of the algorithm can be reduced by running it independently over suitably defined, mutually exclusive layers (see Section 2.3).

In theory, the above algorithm is equally applicable for Binomial and Multinomial conditional transition distributions. However, fitting a Multinomial finite mixture in software such as Stan – which we use for the experiments in Section 4 – faces label switching prob-

[4] Note that a top-down approach with hierarchical clustering algorithms, known as divisive hierarchical clustering, is computationally very expensive with complexity typically being quartic or quintic (Roux, 2015).

Algorithm 1: Mixture model selection algorithm for situation clusters

```
Input  : Data $\mathbf{y}$, prior distribution for $\boldsymbol{\theta}_i$ for $1 \leq i \leq n$, prior distribution for $\boldsymbol{\ell}$.
Output: Optimal number of situation clusters, collection of situation clusters.
Set $allocation \leftarrow \emptyset$, $parameters \leftarrow \emptyset$, $score \leftarrow 0$, $indicator \leftarrow 1$ and $k \leftarrow 2$.
while $indicator \neq 0$ do
    Fit the model as described by Equation 3 with $k$ components.
    Set $score_k$ as the log marginal likelihood of the fitted model using bridge sampling.
    if $score_k \geq score$ then
        $score \leftarrow score_k$
        Set $allocation$ as the posterior allocation of each situation to one of the $k$ components as given by Equation 5.
        Set $parameters$ as the mean posterior estimates of the parameters of each of the $k$ components.
        $k \leftarrow k + 1$
    else
        $indicator \leftarrow 0$
return $allocation, parameters$
```

lems among the components which can results in identifiability issues (Frühwirth-Schnatter, 2006; Mena and Walker, 2015). This is beyond the scope of this paper, and the subject of further research. Section 4 presents experiments for the Binomial case. We discuss possible approaches for circumventing the identifiability issues for the Multinomial finite mixture in Section 5.

3.2.2 Identifying the Edge Clusters

Identifying the edge clusters in an event tree requires a modification to the standard finite mixture modelling problem. Consider an event tree $\mathcal{T}$ with n edges which can all potentially be in the same edge cluster. For edge $e_i \in E(\mathcal{T})$, let $H(e_i)$ denote the conditional holding time random variable, for $1 \leq i \leq n$. Let $\mathbf{y}_i = \{y_{i1}, y_{i2}, \ldots, y_{in_i}\}$ where n_i indicates the number of individuals who traverse edge e_i in our random sample and y_{ij} represents the observed holding time for the jth individual traversing this edge, for $1 \leq i \leq n$ and $1 \leq j \leq n_i$. Let $\mathbf{y} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n\}$ be the data vector and $\boldsymbol{\pi} = \{\boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \ldots, \boldsymbol{\pi}_n\}$ be the parameter vector where $\boldsymbol{\pi}_i$ denotes the parameters associated with the conditional holding time distribution on edge e_i. Similar to the situation clusters in Section 3.2.1, the model selection problem here can be described as identifying the number and composition of the edge clusters in $\mathcal{T}$. However, in this case, we fit a mixture model for each data point y_i instead. This gives us

$$p(\mathbf{y}) = \prod_{i=1}^{n} \sum_{k=1}^{K} \ell_k \prod_{j=1}^{n_i} p(y_{ij} \,|\, \pi_k). \tag{6}$$

The pseudo-code for this algorithm is identical to the pseudo-code in Algorithm 1 with the exceptions that the adapted mixture model to be fit is given by Equation 6 and the posterior allocation is calculated as below

$$z_i^* = \underset{k \in \{1,2,\ldots,K\}}{\arg\max}\; p(z_i^k = 1 \,|\, \mathbf{y}_i), \tag{7}$$

$$\text{where} \quad p(z_i^k = 1 | \mathbf{y}_i) = \frac{\ell_k \prod_{j=1}^{n_i} p(y_{ij} \,|\, \pi_k)}{\sum_{m=1}^{K} \ell_m \prod_{j=1}^{n_i} p(y_{ij} \,|\, \pi_m)}.$$

4 Experiments

In this section, we perform a series of computational experiments on simulated data to demonstrate the performance and properties of our proposed mixture modeling approach to model selection in CEGs. Similar to Section 3.2, we consider the cases of identifying situation clusters and edge clusters separately. Throughout this section, we will use 'stages/staging' to refer to the ground-truth clusters among the situations and edges, and 'clusters/clustering' to refer to the clustering obtained by an algorithm. The experiments described in this section were run in R using the RStudio IDE on a 1.6 GHz MacBook Air with 8GB memory and were parallelized to run on 4 cores. The code for the experiments is provided as part of the supplementary materials.

4.1 Situation Clusters

Here, we compare the performance of our proposed methodology for identifying situation clusters, described in Section 3.2.1, to that of the AHC algorithm. We shall only consider the Binomial case, i.e. where the conditional transition probabilities for the situations follow a Binomial distribution. We simulate 400 datasets for eight different scenarios (50 for each scenario) by setting the number of situations as 50, 200 or 450, and the number of generating stages as 2, 4 or 7 with the exception of the scenario with 50 situations and 7 stages. Each situation has a total of 250 observations. We do not consider the case of 50 situations and 7 stages as this results in some stages having very few data points which realistically makes it extremely difficult to identify the 7 stages correctly for any algorithm. While generating the datasets, the underlying Binomial success probabilities for the various stages are chosen to be distinct enough to minimise issues relating to identifiability (Frühwirth-Schnatter, 2006; Mena and Walker, 2015). Further, the number of situations belonging to the different stages is chosen at random for each simulation whilst ensuring that no stage has fewer than two situations.

During each of the 50 simulations, for each of the eight datasets corresponding to the eight scenarios, we run the AHC algorithm, and the mixture modelling approach in Algorithm 1 in Stan using the dataset.

For the clustering obtained by AHC, we record the number of clusters, time taken (as clock-time in seconds) to run the algorithm and two measures of accuracy of the clustering compared to the ground-truth staging, namely, the normalised mutual information (NMI) score and the Rand index. The NMI score and the Rand index (see Appendix A for more information) assesses the accuracy of the clustering labels compared to the ground-truth labels; for both of these, a score of 0 indicates poor clustering accuracy and 1 indicates perfect clustering. Recall here that the AHC algorithm begins by considering the finest partition where it treats each situation as a singleton cluster, and returns a hard clustering on the situations.

To obtain the clustering from the mixture modelling approach (as described in Algorithm 1), we fit two or more Binomial mixture models in Stan. For fitting a Binomial mixture model in Stan, we run 4 chains, each with 1000 warmup iterations and 2000 post-warmup iterations. For the clustering obtained by the mixture modelling approach, we record the number of clusters, time taken (as clock-time in seconds), the NMI score, and the Rand index. Note here that unlike the AHC algorithm, the mixture model clustering begins with the coarsest partition, and returns a soft clustering on the situations through their posterior allocation probabilities. However, as described in Equation 5, we choose a hard allocation of each situation to a single cluster.

The summary of the results is presented in Table 1. Each scenario is defined by the number of situations (reported as # Situations) and the number of underlying stages (reported as # Stages). The number of clusters (reported as # Clusters and the standard deviation of the number of clusters), time taken, NMI score and Rand index are averaged over the 50 simulations for each scenario. In most cases, the mixture model clustering takes a considerably longer time than the AHC, with the exception of the scenario with 450 situations and 2 underlying stages. This occurs due to the difference in the AHC's top-down approach and the mixture model's bottom-up approach. However, the mixture model clustering consistently performs better than the AHC in terms of the clustering ac-

# Situations	# Stages	# Clusters (std dev)		Time Taken		NMI Score		Rand Index	
		AHC	MM	AHC	MM	AHC	MM	AHC	MM
50	2	4.52 (0.707)	2.00 (0)	3.60	70.29	0.59	0.96	0.61	0.98
50	4	6.20 (0.700)	3.50 (0.678)	3.11	159.13	0.67	0.78	0.78	0.84
200	2	7.80 (0.808)	2.08 (0.274)	101.07	227.23	0.49	0.98	0.49	0.99
200	4	10.62 (0.923)	3.90 (0.931)	99.70	780.36	0.57	0.75	0.72	0.84
200	7	14.80 (1.20)	5.82 (0.774)	97.00	1235.65	0.72	0.87	0.84	0.93
450	2	10.52 (1.45)	2.08 (0.279)	1144.17	450.37	0.45	0.98	0.45	0.98
450	4	14.64 (1.47)	4.52 (1.07)	1134.43	3036.13	0.55	0.74	0.71	0.84
450	7	20.08 (1.34)	6.20 (1.14)	1121.28	4039.94	0.68	0.85	0.82	0.92

Table 1: Summary of the results for clustering situations with underlying Binomial conditional transition distributions using the AHC algorithm and the mixture modelling (MM) approach.

curacy metrics. The summary of the convergence results for the simulations is presented in Appendix B.

4.2 Edge Clusters

For the edge clusters, we analyse the performance of our mixture modelling approach in the case where we do not have conjugacy. In this case, the AHC algorithm as described in Section 2.3 is not applicable and hence, we cannot use it for comparative purposes. Here, the conditional holding time data for each edge is assumed to come from a Weibull distribution with known scale parameter and unknown shape parameter. Recall that the Weibull distribution only enjoys a conjugate prior for the scale parameter when the shape parameter is known and the scale parameter is unknown. Similar to the setting in Section 4.1, we simulate 400 datasets for eight different scenarios (50 for each scenario) by setting the number of edges as 50, 200 or 450, and the number of generating stages to 2, 4 or 7 with the exception of the scenario with 50 edges and 7 stages. We generate 30 holding times for each edge. We set the scale parameters for all stages to be 50 and set the underlying shape parameters for the different stages to be distinct enough to minimise identifiability issues. The number of edges belonging to the different stages is chosen at random for each simulation whilst ensuring that no stage has fewer than five edges.

For each clustering obtained by the mixture modelling approach (as described in Section 3.2.2), we fit a Weibull mixture model in Stan with known scale parameter and estimate the unknown shape parameter. We fit the model using 4 chains, each with 1000 warmup iterations and 2000 post-warmup iterations. We record the number of clusters, time taken (as clock-time in seconds), the NMI score, and the Rand index for the clustering obtained through the approach. We enforce a hard clustering as described in Equation 7. The summarised results averaged over the 50 simulations for each of the eight scenarios are presented in Table 2. This approach has very good performance as evidenced by average

# Edges	# Stages	# Clusters (std dev)	Time Taken	NMI Score	Rand Index
50	2	2.00 (0)	133.01	1.00	1.00
50	4	4.00 (0)	353.11	0.99	1.00
200	2	2.00 (0)	676.19	1.00	1.00
200	4	4.04 (0.198)	2034.83	0.99	1.00
200	7	6.28 (1.74)	4623.07	0.88	0.93
450	2	2.00 (0)	1879.68	1.00	1.00
450	4	4.02 (0.141)	9260.44	0.99	1.00
450	7	5.32 (1.92)	11961.49	0.81	0.87

Table 2: Summary of the results for clustering edges with underlying Weibull conditional holding time distributions with known scale parameters and unknown shape parameters using the mixture modelling approach.

values of the number of clusters (reported as # Clusters and the standard deviation of the number of clusters), NMI score and Rand index for all eight scenarios, in particular when the underlying stages are 2 or 4. As for the situation clusters, the summary of the convergence results is presented in Appendix B.

4.3 A Real-World Example

We now present a practical application of our mixture model methodology. This real-world example is based on an intervention to reduce falls-related injuries among the elderly, described originally in Eldridge et al. (2005) and embellished with holding times in Shenvi (2021). Here, we use a simulated dataset on a subset of the intervention process to illustrate how our methodology may be used in practice.

We consider 325 individuals within the community who are assessed to have “High” or “Low” risk of falling. As part of the intervention, a proportion of those who are high-risk

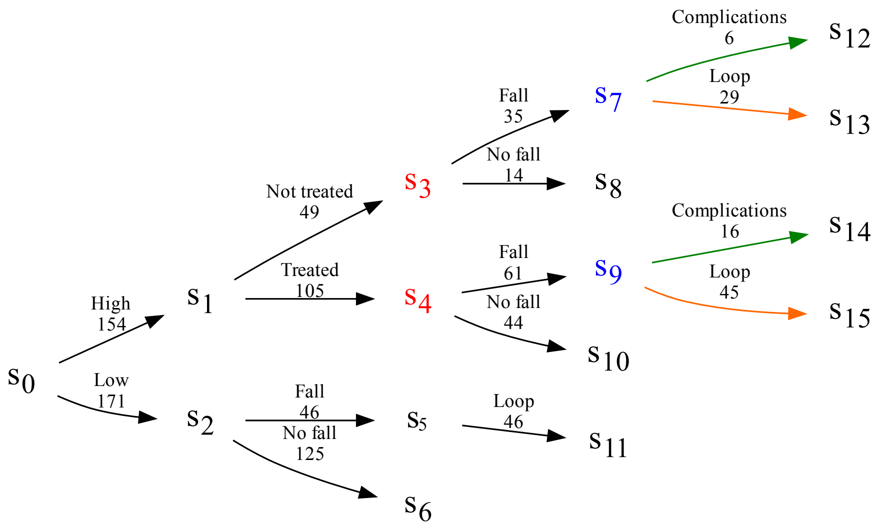


Figure 3: Staged tree for the falls intervention with edge-counts shown along with the edge labels. The situation clusters are denoted by coloured nodes and edge clusters are denoted by coloured edges.

are given treatment in a falls clinic. We assume that a low-risk individual who falls is settled back within the community. Unless they have serious complications, we assume the same holds for a high-risk individual who falls. We are interested in analysing the effects of the intervention on the timing and probability of the falls and eventual outcomes.

Random variables	Description
$H(e_{3,7}), H(e_{4,9})$	Duration to experiencing a fall.
$H(e_{7,12}), H(e_{9,14})$	Duration of post-fall care for suffering serious complications.
$H(e_{7,13}), H(e_{9,15})$	Duration of post-fall care until return to community living.

Table 3: Definition of random variables in Figure 3. $H(e_{i,j})$ refers to the holding time along edge e_{ij} from situations s_i to s_j.

Figure 3 shows the underlying staged tree for the falls-intervention and Table 3 gives the edges which have associated holding times. In the figure, nodes and edges which are coloured black are in their own singleton cluster. Observe situations s_3 and s_4 that share the same node colouring but their edges with holding times (i.e. edges $e_{3,7}$ and $e_{4,9}$) are both coloured black. This can be interpreted as the treatment does not affect the probability of falling for high-risk individuals, but it affects the duration to experiencing a fall. Further, we see that situations s_7 and s_9 not only share the same node colouring but also their corresponding holding times edges are coloured the same. This indicates that post suffering a fall, the outcomes and timing of the outcomes are the same for high-risk individuals who are treated and those who are not. By the definition of a stage, s_7 and s_9 are in the same stage but s_3 and s_4 are not.

We fit Binomial mixture models for s_2, s_3 and s_4, and then separately for s_7 and s_9. For the situations, the results obtained using the CEG model selection approach based on mixture models are presented in Table 4. We can see that the approach found the right number of generating components, and the estimated parameters are close to the generating parameters. Results obtained with AHC are very similar but not presented.

Situation	Generating parameters	Estimated parameters
s_2	0.28	0.272
s_3 & s_4	0.60	0.622
s_7 & s_9	0.20	0.203

Table 4: Generating and estimated parameters of success for the Binomial distributions.

For the edges we fit Weibull models. Here we had a different shape and scale parameter for each generating component, and also the number of observations for each edge was not the same as in the experiments (the number of observations is the edge count for a given edge). AHC cannot be used for this fit. However, we can customise the mixture model and define problem-specific priors. We were able to learn both parameters of the Weibull distribution. We did this for edges $e_{3,7}$, $e_{4,9}$ and $e_{2,5}$, and separately for $e_{7,12}$, $e_{7,13}$, $e_{9,14}$, $e_{9,15}$ and $e_{5,11}$. The number of generating components were correctly retrieved, and the estimated parameters were close to the parameters of the generating Weibull distributions as shown in Table 5.

Edge	Generating shape, scale parameters	Estimated shape, scale parameters
$e_{3,7}$	2, 220	2.986, 269.037
$e_{4,9}$	5, 300	4.348, 301.203
$e_{2,5}$	15, 350	15, 347.349
$e_{7,12}$ & $e_{9,14}$	10, 50	8.306, 48.107
$e_{7,13}$ & $e_{9,15}$	3, 25	4.259, 24.396
$e_{5,11}$	2, 10	2.317, 11.142

Table 5: Generating and estimated parameters for the Weibull distributions.

5 Discussion

In this paper we have shown that by viewing model selection for CEGs as a clustering problem, we can use a mixture modelling approach for model selection in CEGs. We demonstrated that this approach is very promising when the conditional holding time distributions do not have conjugate priors and also for robustly scaling to a larger number of situations (or equivalently, edges) as compared to the AHC algorithm even under the assumption of conjugacy.

This work opens up several avenues for future work; most excitingly for new applications of CEGs for processes with arbitrary holding time distributions and/or a large number of nodes in its event trees. Further, the soft clustering provided naturally by the mixture modelling approach can be used with a Bayesian model averaging setting such as in Strong and Smith (2022a) for robust explanatory analyses using a set of top-scoring models rather than just the maximum *a posteriori* model.

There are challenges that will require further study. The conditional probability distribution for a situation with three or more emanating edges follows a Multinomial distribution. Fitting a Multinomial mixture model in Stan faces identifiability issues. Betancourt (2017) recommends identifying degenerate Bayesian mixture models by either using non-exchangeable priors or enforcing an ordering on the parameters. In the Binomial case, we used the latter approach on the probability of success parameter of the Binomial distribution. However, for the Multinomial case, enforcing an ordering is not sufficient as a Multinomial with k categories has degree of freedom $k-1$ and it is not straightforward how to enforce an ordering on all $k-1$ categories at once. There are two possible approaches that we could consider for further study. The first is that a Multinomial distribution can be written as a series of consecutive Binomial distributions (Strong and Smith, 2022b), and the second is that for a specific application, non-exchangeable priors could be used (as done in Section 4.3 for the Weibull distribution).

Further, our current approach scales well in the number of situations or edges in the tree, but estimating the number of components using the current method is not easily scalable as the number of underlying stages increases. Within a specific application, in order to minimise the computational load, it is advisable to elicit a suitable range for the number of components prior to commencing the model selection process.

Despite these challenges that require further study, the approach that we propose in this paper vastly extends the applicability of CEGs and will open up a range of opportunities.

Appendix

A Clustering Accuracy Metrics

The normalized mutual information (NMI) score and the Rand index are two popular metrics for comparing the accuracy of a clustering algorithm. Let GT denote the ground-truth or generating cluster labels of the data points and $Pred$ denote their corresponding predicted cluster labels. The NMI is a normalization of the mutual information score and it is obtained as follows:

$$NMI(Pred, GT) = \frac{I(Pred, GT)}{\sqrt{H(Pred)H(GT)}},$$

where $I(Pred, GT)$ denotes the mutual information between the two labellings and $H(Pred)$ denotes the entropy of $Pred$. Here, a score of 0 indicates no mutual information whereas a score of 1 indicates perfect correlation.

The Rand index measures the percentage of correct decisions made by the clustering algorithm and is given as

$$RI(Pred, GT) = \frac{TP + TN}{TP + FP + TN + FN},$$

where TP, TN, FP and FN are the true positives, true negatives, false positives and false negatives respectively in $Pred$ compared to GT. The range of the Rand index is $[0, 1]$ with a higher value indicating a better clustering accuracy.

B Convergence Results for the Experiments

B.1 Situation Clusters

Unlike the AHC algorithm which uses closed form equations to estimate the parameters of interest, the mixture model clustering implemented in Stan estimates the parameters of interest using a No-U-Turn Sampler (NUTS) (Carpenter et al., 2017). Hence, as a diagnostic check we analyze whether the parameters relating to the Binomial distribution

# Situations	# Stages	Prop Converging Lvl1	Prop Converging Lvl2
50	2	0.96	0.96
50	4	0.83	0.95
200	2	0.90	0.90
200	4	0.43	0.58
200	7	0.68	0.75
450	2	0.92	0.94
450	4	0.36	0.53
450	7	0.50	0.63

Table 6: Summary of the convergence results for clustering the situations using the mixture modelling approach.

for each component converge. The computation is said to converge when the split-$\hat{R} < 1.01$ as recommended by Vehtari et al. (2021). This is a much tighter bound compared to the original recommended bound of 1.10 (Gelman and Rubin, 1992). For comparative purposes, we also check convergence under the 1.10 threshold. Table 6 shows the proportion of Binomial parameters that converged at the threshold of 1.01 (reported as Prop Converging Lvl1) and 1.10 (reported as Prop Converging Lvl2) for each scenario averaged over the 50 simulations. Over two-thirds of the parameters converged under both thresholds for each scenario except for the scenarios of 200 situations & 4 stages and 450 situations & 4 or 7 stages.

B.2 Edge Clusters

We analyse the convergence properties of the mixture modelling approach to clustering the edges under the thresholds of 1.01 and 1.10 for the split-$\hat{R}$. This is summarised in Table 7 with the threshold of 1.01 reported as Prop Converging Lvl1 and that of 1.10 reported

# Edges	# Stages	Prop Converging Lvl1	Prop Converging Lvl2
50	2	1.00	1.00
50	4	1.00	1.00
200	2	1.00	1.00
200	4	0.98	1.00
200	7	0.56	0.67
450	2	1.00	1.00
450	4	0.99	1.00
450	7	0.26	0.34

Table 7: Summary of the convergence results for clustering the edges using the mixture modelling approach.

as Prop Converging Lvl2. Almost all 50 simulations converged for each scenario, with the exception of 200 edges & 7 stages and 450 edges & 7 stages.

The convergence results when we have 7 stages are not as good as for fewer stages. Recall that we compared models using their log marginal likelihoods which were approximated using bridge sampling. This is equivalent to using the Bayes Factor (Kass and Raftery, 1995) where all the models are *a priori* equally likely. The Bayes Factor, whilst an extremely common approach to model comparison, has several drawbacks. It is very sensitive to priors and difficult to approximate accurately (Gronau et al., 2020; Schad et al., 2022; Oelrich et al., 2020). Therefore, the approximated Bayes Factor is not always suitable for comparing models especially when the approximation is carried out in a black-box manner. Further, observe that when the number of estimated components is two, we only estimate two log marginal likelihoods and make one Bayes Factor comparison. However, when the estimated number of components is 7, we have 6 log marginal likelihood estimations and 5 Bayes Factor comparisons; thereby increasing the possibility of errors caused due to the use of Bayes Factors. In practice, we recommend careful checks of the Stan and bridgesampling outputs, and the use of post-hoc analysis if necessary; see Section 5.

SUPPLEMENTARY MATERIAL

R Code for the Situation Clusters: Zip file containing the R code files and Stan file for the simulations in Section 4.1 of the main article as well as the summary of the clustering of the simulated data. (Zipped file titled 'Situation Clusters.zip')

R Code for the Edge Clusters: Zip file containing the R code files and Stan file for the simulations in Section 4.2 of the main article as well as the summary of the clustering of the simulated data. (Zipped file titled 'Edge Clusters.zip')

References

Barclay, L. M., Collazo, R. A., Smith, J. Q., Thwaites, P. A., and Nicholson, A. E. (2015). The dynamic chain event graph. *Electronic Journal of Statistics*, 9(2):2130–2169.

Berger, J. O. and Jefferys, W. H. (1992). The application of robust Bayesian analysis to hypothesis testing and Occam's razor. *Journal of the Italian Statistical Society*, 1(1):17–32.

Betancourt, M. (2017). Identifying Bayesian mixture models.

Boutilier, C., Friedman, N., Goldszmidt, M., and Koller, D. (1996). Context-specific independence in Bayesian networks. In *Proceedings of the Twelfth International Conference on Uncertainty in Artificial Intelligence*, pages 115–123.

Carli, F., Leonelli, M., Riccomagno, E., and Varando, G. (2022). The R Package stagedtrees for Structural Learning of Stratified Staged Trees. *J. of Stat. Soft.*, 102:1–30.

Carpenter, B., Gelman, A., Hoffman, M. D., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M., Guo, J., Li, P., and Riddell, A. (2017). Stan: A probabilistic programming language. *Journal of Statistical Software*, 76(1):1–32.

Collazo, R. A., Görgen, C., and Smith, J. Q. (2018). *Chain event graphs*. CRC Press.

Collazo, R. A. and Smith, J. Q. (2018). An N time-slice dynamic chain event graph. *arXiv:1808.05726*.

Cowell, R. G. and Smith, J. Q. (2014). Causal discovery through MAP selection of stratified chain event graphs. *Electronic Journal of Statistics*, 8(1):965–997.

Eldridge, S., Spencer, A., Cryer, C., Parsons, S., Underwood, M., and Feder, G. (2005). Why modelling a complex intervention is an important precursor to trial design: lessons from studying an intervention to reduce falls-related injuries in older people. *Journal of Health Services Research & Policy*, 10(3):133–142.

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

Frühwirth-Schnatter, S. (2006). *Finite mixture and Markov switching models*. Springer.

Gelman, A. and Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences. *Statistical Science*, 7(4):457–472.

Gronau, Q. F., Sarafoglou, A., Matzke, D., Ly, A., Boehm, U., Marsman, M., Leslie, D. S., Forster, J. J., Wagenmakers, E.-J., and Steingroever, H. (2017). A tutorial on bridge sampling. *Journal of Mathematical Psychology*, 81:80–97.

Gronau, Q. F., Singmann, H., and Wagenmakers, E.-J. (2020). bridgesampling: An R package for estimating normalizing constants. *Journal of Statistical Software*, 92(10):1–29.

Jabbari, F., Visweswaran, S., and Cooper, G. F. (2018). Instance-specific Bayesian network structure learning. In *Proceedings of the Ninth International Conference on Probabilistic Graphical Models*, pages 169–180. PMLR.

Kass, R. E. and Raftery, A. E. (1995). Bayes factors. *Journal of the American Statistical Association*, 90(430):773–795.

Leonelli, M. and Varando, G. (2022). Highly efficient structural learning of sparse staged trees. In *Proceedings of the Eleventh International Conference on Probabilistic Graphical Models*. PMLR.

Mena, R. H. and Walker, S. G. (2015). On the Bayesian mixture model and identifiability. *Journal of Computational and Graphical Statistics*, 24(4):1155–1169.

Minka, T. P. (2003). Bayesian inference, entropy, and the multinomial distribution. Technical report, Microsoft Research.

Nielsen, F. (2016). Hierarchical clustering. In *Introduction to HPC with MPI for Data Science*, pages 195–211. Springer.

Oelrich, O., Ding, S., Magnusson, M., Vehtari, A., and Villani, M. (2020). When are Bayesian model probabilities overconfident? *arXiv:2003.04026*.

Poole, D. and Zhang, N. L. (2003). Exploiting contextual independence in probabilistic inference. *Journal of Artificial Intelligence Research*, 18:263–313.

Roux, M. (2015). A comparative study of divisive hierarchical clustering algorithms. *arXiv:1506.08977*.

Schad, D. J., Nicenboim, B., Bürkner, P.-C., Betancourt, M., and Vasishth, S. (2022). Workflow techniques for the robust use of Bayes factors. *Psychological Methods*.

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

Shenvi, A. (2021). *Non-Stratified Chain Event Graphs: Dynamic Variants, Inference and Applications*. PhD thesis, The University of Warwick.

Shenvi, A. and Smith, J. Q. (2019). A Bayesian dynamic graphical model for recurrent events in public health. *arXiv:1811.08872*.

Shenvi, A. and Smith, J. Q. (2020). Constructing a chain event graph from a staged tree. In *Proceedings of the Tenth International Conference on Probabilistic Graphical Models*. PMLR.

Shenvi, A., Smith, J. Q., Walton, R., and Eldridge, S. (2018). Modelling with non-stratified chain event graphs. In *International Conference on Bayesian Statistics in Action*, pages 155–163. Springer.

Silander, T. and Leong, T.-Y. (2013). A dynamic programming algorithm for learning chain event graphs. In *International Conference on Discovery Science*, pages 201–216. Springer.

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

Strong, P. and Smith, J. Q. (2022a). Bayesian model averaging of chain event graphs for robust explanatory modelling. In *Proceedings of the Eleventh International Conference on Probabilistic Graphical Models*. PMLR.

Strong, P. and Smith, J. Q. (2022b). Scalable model selection for staged trees: Mean-posterior clustering and binary trees. In *International Conference on Bayesian Statistics in Action*, pages 23–34. Springer.

Vehtari, A., Gelman, A., Simpson, D., Carpenter, B., and Bürkner, P.-C. (2021). Rank-normalization, folding, and localization: An improved $\hat{R}$ for assessing convergence of MCMC. *Bayesian Analysis*, 1(1):1–28.