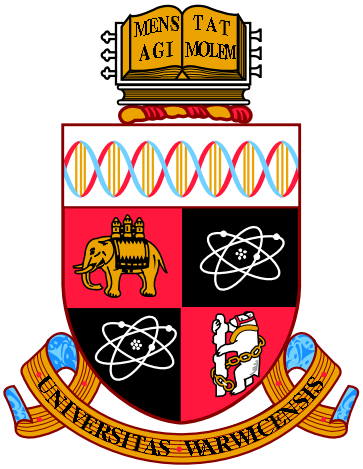


Methodological Advances in Explainable Modelling using Chain Event Graphs

by

Peter Strong

Thesis

Submitted to the University of Warwick

for the degree of

Doctor of Philosophy

in

Mathematics of Systems

Mathematics for Real-World Systems CDT

April 2023

WARWICK
THE UNIVERSITY OF WARWICK

Contents

List of Tables

List of Figures

Acknowledgments

Firstly, this research would not have been possible without Jim Q. Smith (a.k.a Jimmy the Fish), for being encouraging, knowledgeable and enthusiastic. His expertise, insights, and feedback have been critical in shaping my research and helping me develop as a researcher. Thank you for all your support: you have given me the confidence to pursue ambitious ideas and take risks in my work. I feel privileged to have worked with you and you have been instrumental to the completion of this thesis.

I would also like to thank my collaborators for projects I worked on during my thesis for their invaluable contributions. In particular, Aditi, Alys, Connor, Gareth, and Nadia have supported and guided me throughout my research; your ideas have been pivotal in shaping the outcome of my work. I am also grateful to Jack for his valuable comments.

A big thank you to everyone in the MathSys CDT for providing a caring and friendly research environment and in particular, the members of my CDT: Alex, Charlie, Emma, George, Kendal, Matt, Paul, Ricky, Stas, Susie, and Yuanyi. Although they were not directly involved in my research, their support and camaraderie have been invaluable.

I am grateful to the Engineering and Physical Sciences Research Council (EPSRC) and the Medical Research Council for their funding support, which made my research possible.

I am thankful for my parents and sister's constant support and encouragement and for everything they have done for me.

Finally, I would like to express my deepest gratitude to my partner, Koura,

for her unwavering support throughout this entire process. Thank you for listening to my rambles on potential research ideas for chain event graphs for the last few years and helping me present my thoughts in a legible manner. This work would not have been possible without you.

Declarations

I declare that the work presented in this thesis is my own except when stated otherwise. This thesis has not been submitted in this form or similar for examination to any other institution. Some of this work has been published or is currently in the submission process as described below.

Chapter 3 includes work based on software that I was involved in developing. This has led to a pre-print currently under review entitled "cegpy: Modelling with Chain Event Graphs in Python" for which I am a contributing author. The computational experiments contained in this thesis are done using this package and extensions that I have developed from the base software.

Chapter 4 is partially based on material in a published paper in which I am the lead author that appeared in the proceedings of the 11th International Conference on Probabilistic Graphical Models. This paper is titled "Bayesian Model Averaging of Chain Event Graphs for Robust Explanatory Modelling". This is joint work with Jim Q. Smith.

Chapter 5 is based on a paper to appear in the proceedings of Bayesian Young Statisticians Meeting 2022, in which I am the lead author. This pre-print is entitled "Scalable Model Selection for Staged Trees: Mean-posterior Clustering and Binary Trees" and is joint work with Jim Q. Smith.

Chapter 6 is based on material in a published paper appearing in the proceedings of the Bayesian Young Statisticians Meeting 2021, for which I am the lead author. The paper is entitled "Towards a Bayesian Analysis of Migration Pathways Using Chain Event Graphs of Agent Based Models". This is joint work with Jim Q. Smith and Alys McAlpine.

There were several substantial pieces of research I was involved with over the period of this PhD which informed this work but are not central to it. I have not reported details of these here in order to maintain the coherence of the subject matter and to keep the story tight and without distractions.

- I led on the publication of a Bayesian decision analysis paper designed to guide the balancing competing objectives in COVID. This has now appeared in the Journal of the Operational Research Society. "Building A Bayesian Decision Support System for Evaluating COVID-19 Countermeasure Strategies" [Strong et al., 2021]

- I am second author on a paper currently under submission that "On Bayesian Dirichlet Scores for Staged Trees and Chain Event Graphs" [Hughes et al., 2022]. This work defined the first score-equivalent scoring function for staged trees. This scoring function is used in Chapter 5.

- I am also second author on a report titled "A Dynamic Graphical Model of Illicit Drug Production"; this used flow graphs to provide decision support about elicit drug production.

- Finally, I am a coauthor for a mixed methods study on the use of a website that provided information to carers to help with their caring responsibility. This led to a paper currently under review "Care Companion: a mixed methods, real world evaluation of the use of an online resource to support informal carers" [Dale et al., 2022].

Abstract

Chain Event Graphs (CEGs) are an easily interpretable, versatile class of probabilistic graphical models that represent context-specific relationships and asymmetric event unfoldings. As an asymmetric extension of discrete Bayesian networks, CEGs provide a compact illustration of detailed dependence structures through the use of colour and by modifying the graphs topology.

Although other model selection methods have been studied, CEG model selection literature has primarily focused on obtaining the maximum a posteriori (MAP) CEG. However, this method ignores model uncertainty and therefore the uncertainty of their contained independence statements. We propose using Bayesian model averaging (BMA) to quantify model uncertainty, leading to more robust inference by comparing features across high-scoring models. We provide a simple modification of an existing model selection algorithm, that samples the model space, to illustrate the efficacy of Bayesian model averaging compared to more standard MAP modelling.

Recent improvements in structure learning have not mitigated the computational complexity involved in modelling larger applications. They either: fail to scale efficiently when the number of events considered increases; do not find comparable models to existing methods or *a priori* restrict the model space. We propose an alternative algorithm, using a totally-ordered hyperstage, to obtain a quadratically-scaling structural learning algorithm for staged trees, restricting the model space *a-posteriori*. Our approach outperforms existing methods in computational time, whilst providing comparable model scores. This enables learning more complex relationships than existing model selection techniques by expanding the model space.

We consider how CEGs can improve the explainability of Agent-Based Models (ABMs), a popular model class in social science, by providing a Bayesian framework. Although ABMs lack the methods to embed more principled strategies of performing inference to estimate and validate the models, CEGs can fill this gap by accurately representing ABMs. Using a CEG, we illustrate transforming an elicited ABM into a Bayesian framework and outline the benefits of this approach.

Abbreviations

ABM	Agent-Based Model
AHC	Agglomerative Hierarchical Clustering
BCEG	Binary Chain Event Graph
BD	Bayesian Dirichlet
BDeu	Bayesian Dirichlet equivalent uniform
BDepu	Bayesian Dirichlet equivalent path uniform
BF	Bayes Factor
BMA	Bayesian Model Averaging
BN	Bayesian Network
CEG	Chain Event Graph
CHDS	Christchurch Health and Development study
CT-DCEG	Continuous Time Dynamic Chain Event Graph
DAG	Directed Acyclic Graph
DCEG	Dynamic Chain Event Graph
IDSS	Integrated Decision Support System
MAP	*Maximum a posteriori*
MyTh MAP-IN	Myanmar-Thailand Migration Planning and Intermediary Networks
PGM	Probabilistic Graphical Model

List of Symbols

G	A graph
$V(G)$	The vertices of a graph
$E(G)$	The edges of a graph
$\mathcal{T}$	A tree
$L(\mathcal{T})$	Leaves in a tree
$S(\mathcal{T})$	Situations in a tree
$ch(v)$	Children of vertex
$F(s)$	Floret of a situation
$\Lambda(\mathcal{T})$	The set of all root-to-leaf paths in a tree
$\Phi_{\mathcal{T}} = \{\boldsymbol{\theta_v} \vert v \in S(\mathcal{T})\}$	Set of conditional transition probability parameters
$\mathcal{S}$	A staged tree
$\mathcal{C}$	Chain Event Graph
$K(\mathcal{S})$	Set of positions in a staged tree
X_i	Variable i
$\mathcal{X}$	Vector of variables
$\mathcal{X}^k$	Vector of variables up to variable k
$\mathbb{U}$	Stages of a CEG
$\boldsymbol{Y}$	A random sample with no missing data
K	Number of stages
k_i	number of outgoing edges of stage i
$\boldsymbol{\alpha_i}$	Dirichlet prior parameter vector
$\Gamma(z)$	Gamma function

$\bar{\boldsymbol{\alpha}}$	Sum of elements in a vector α
$\boldsymbol{\alpha}_i^*$	Sum of Dirichlet prior parameter and data
$Q(\mathcal{C})$	The marginal likelihood of CEG $\mathcal{C}$
$g(x)$	Log of the Gamma function
μ_{ij}	Prior means of the Dirichlet parameter
$\bar{\boldsymbol{\alpha}}_0$	Effective sample size at the root
$\boldsymbol{H}$	Hyperstage
$\#S$	Number of elements in a set S
$B(m)$	mth Bell number
$u_{i\oplus j}$	The stage obtained by combining stages u_i and u_j
M_k	Model k
M	Number of models in the model space
$BF(M_k, M_l)$	BF of two models
β	Parameter of cut off point for Occam's window
$\mathcal{S}'$	Set of models below the β cut off point
$\mathcal{R}$	Set of models removed by the Occam's razor
$\hat{\mathcal{S}}$	Set of well-performing models
$S_{i,j}$	Staging of hyperset i in model j
$\hat{\boldsymbol{S}}_n$	Set of independent samples from the model space
$U(\boldsymbol{S})$	Set of unique elements in a vector $\boldsymbol{S}$
$\hat{\boldsymbol{\pi}}_n$	Approximation of BMA weights
$\boldsymbol{\pi}$	BMA weights
$\boldsymbol{\pi}^*$	Occam's window BMA weights
γ	Parameter for the stopping condition in wr-HAC
R	Parameter for setting number of runs of wr-HAC
N	Number of situations in the model space
$\boldsymbol{\beta}_i$	Uniform model prior counts

Chapter 1

Introduction

1.1 Motivation

Graphical models are used by researchers from a wide variety of domain areas to understand the structural relationships and interactions between different variables and decisions. Combining probability theory and graph theory, the result is a visual representation, capable of providing intuitive and comprehensive explanation of a system whilst also representing highly complex relationships between variables in a cohesive manner.

A key benefit of graphical models is that they can often be explained to stakeholders and researchers from non-mathematical domains with little statistical training; as such, they are a compelling communication tool and are regularly employed in interdisciplinary research.

Probabilistic Graphical Models (PGMs) can provide a visually compelling and intuitively intelligible representation of the probabilistic associations encoded within its statistical model: PGMs enable statisticians to understand complex relationships between variables through their representation and topology. The graphical nature of the model means that this can be done visually, without any requirement to further investigate into the statistical model's parameters.

Of these graphical models, one class– the Bayesian Network (BN)– has enjoyed success across many domain areas, including agriculture [Drury et al., 2017], metereology [Cano et al., 2004], antiterrorism [Hudson et al., 2005] and ecology [Rigosi et al., 2015].

BNs are ideal for modelling certain contexts, when an expert focuses on the relationships between a pre-defined set of variables [Collazo et al., 2018]. However, despite their flexibility and wide use, BNs have some widely-documented drawbacks

which limit their suitability. When experts elicit judgements regarding variables, they often describe them as *asymmetric processes*, where context and previous history of a unit is key to describing the outcomes of events. We note that BNs do not necessarily describe asymmetric events coherently; instead of considering variables as part of a process, they consider them as existing in a product structure, which can lead to model descriptions which are inaccurate and border on the absurd.

Example 1 (Asymmetry in BNs) *When completing a medical questionnaire, a participant may be asked how regularly they drink alcohol. In an asymmetric process, a participant reporting "never" would not be asked "When you drink, how many units do you drink?"; this question does not make sense to ask. Modelling this as a symmetric process may use the follow-up question asking the participant to state the number of units drunk when they do drink – with a minimum number of one – as a variable. BNs model processes as symmetric and require a response for every individual for every variable. This is not to say that strategies to deal with missing data in BNs do not exist. However in instances like this assuming it is missing at random or to impute such values further obfuscates the problem, leading to inaccurate information and a poor representation.*

In addition, BNs cannot fully illustrate context-specific independence statements, where independences only hold for specific values in specific circumstances [Spiegelhalter and Lauritzen, 1990]. Although various extensions and adaptations– such as context-specific and object-oriented BNs– have been made to the BN class to expand their suitability, BNs still have no capacity to accommodate asymmetries within the graphical description of the model. Therefore, if we are to identify a model which provides explainability in a wider range of circumstances, we must look beyond the Bayesian Network.

This example motivates the use of Chain Event Graphs (CEGs), a class of PGM introduced in Anderson and Smith [2005]. Developed originally from event trees, which provide an intuitive framework to describe an unfolding process [Shafer, 1996], Chain Event Graphs embellish this structure by colouring the nodes based on the distributions over each edge and transforming its graph structure to create a compact representation [Collazo et al., 2018]. The CEG class is hugely expressive [Insua and French, 2010], with the ability to represent complex, context-specific independence relationships across symmetric and asymmetric event spaces.

Compared with a tree, CEGs provide a comparatively compact representation, with each CEG for a given tree providing a different explanation of the underlying process. Therefore, for a given event tree, the model selection depends on

deciding between independence statements through the potential ways of colouring a tree.

Since Anderson and Smith [2005]'s original development, different applications of CEGs have been successfully established such as in causal analysis [Thwaites et al., 2010; Thwaites, 2013; Cowell and Smith, 2014], dynamic variants [Freeman and Smith, 2011b; Barclay et al., 2015; Collazo, 2017; Shenvi, 2021] and for modelling latent processes in hierarchical models [Bunnin et al., 2020; Shenvi et al., 2023].

Obtaining the most likely– maximum a posteriori (MAP)– model has been the focus of much existing model selection literature [Collazo, 2017; Silander and Leong, 2013; Freeman and Smith, 2011a]. This research has focused on identifying a single, best-performing model. However, relying solely on MAP selection ignores the model uncertainty, which can lead to unreliable results. As a result, new methods are needed to quantify the uncertainty of models; Bayesian model averaging offers a more robust strategy for quantifying model uncertainty by identifying shared features across multiple high-scoring models.

Despite recent improvements in structure learning, the computational complexity involved in modelling larger applications remains a challenge. Existing methods either restrict the set of models *a priori*, do not scale efficiently as the number of events considered increases or fail to find comparable models to existing methods. Additionally, when comparing stages with contrasting effective sample sizes, optimal combinations can sometimes appear odd, leading to inappropriate model selection [Collazo and Smith, 2016].

CEGs offer a powerful framework for modelling complex systems and making inferences under uncertainty. However, they are a comparatively new model class and their application by researchers from non-statistical backgrounds has thus far been limited. Agent-Based Models (ABMs)– an egocentric class of models used to simulate real-world systems– are more widespread in the social sciences but their interpretability, estimation and validation can be enhanced by exploiting the relationship of this class to the CEG. By using CEGs to embed a Bayesian framework in ABMs, we can provide a way to capture the causal relationships and dependencies between agents and variables in ABMs, allowing for more principled strategies of performing inference. By providing a method to embellish an ABM into a CEG, there is potential to broaden the use of CEGs into new domains, for use by non-statistical researchers to enhance their understanding of real-world systems.

When using prior information, containing expert judgement, in any Bayesian analysis it is important that it is used consistently. However, prior probabilities and

distributional parameters are often set by default, without reference or consideration for each other. This can lead to an imbalance in how data and expert judgement are treated in model development, where expert elicitation is undervalued and not represented consistently in some models.

With these points in mind, the work presented in this thesis aims to address the following research questions:

1. How can Bayesian model averaging be used to quantify model uncertainty and improve the robustness of inference for CEGs, and how does it compare to existing methods, such as MAP selection?

2. How can the computational complexity of structural learning in larger applications be mitigated, and can an alternative algorithm be proposed and implemented that provides more efficient and accurate results than existing methods?

3. How can CEGs be used to provide a Bayesian framework to make ABMs more explainable and provide more principled strategies for performing inference to estimate and validate the models, and what are the benefits of this approach?

4. When modelling with CEGs, how can we ensure a balanced approach when considering expert judgement and data to maintain consistency?

1.2 Outline of Thesis

The work in this thesis has the following structure.

We begin in Chapter 2 by briefly outlining the fundamentals of graph theory and PGMs as these apply to the material presented in later chapters of this thesis. We then briefly review discrete BNs and highlight their limitations through illustrative examples. We discuss how these limitations motivate the use of CEGs and alternative graphical models.

In Chapter 3, we provide a detailed review of CEGs. This includes the formal definition of staged trees, CEGs and non-stratified CEGs. We then review model selection methods including conjugate learning, the setting of priors and model selection algorithms. This chapter also includes a review of the existing software used to perform model selection for CEGs, demonstrating the need for software able to model non-stratified CEGs. Finally, the details of `cegpy`, a python package I was involved in developing for non-stratified CEGs, is included, in which we provide an demonstration of its functionality.

The proceeding four chapters include the main body of original methodological research that makes up this thesis. Chapter 4 explores the use of Bayesian Model Averaging (BMA) to quantify the confidence that surrounds independence statements learned from data within a CEGs model selection.

The next chapter, Chapter 5, explores the use of restricting the hyperstage for more efficient model selection for CEGs. To do this, we introduce an ordering on each hyperset. We also explore traversing the equivalence class of CEGs in order to obtain a wider class of models able to represent a wider set of relationships between events.

Chapter 6 presents ongoing work on the use of elicited expert judgement. In particular, we consider how to elicit a CEG from an ABM and compare the similarities and differences in modelling approaches between these two methods.

Chapter 7 discusses how prior information can be managed to ensure that expert judgement is reflected consistently in a CEG. Here we motivate and define an invariance condition and show how it can be satisfied to ensure the consistent use of expert judgement and data.

We conclude in Chapter 8, where we summarise the contributions given in this thesis, discuss ongoing work and detail areas for further research.

Chapter 2

Preliminary information

In this chapter, we provide a background of some of the key concepts for this thesis. Section 2.1 details a review of relevant concepts from graph theory. Section 2.2 gives a broad overview of PGMs. This section has a particular focus on BNs, a very popular class of PGMs. Through discussing the limitations of BNs we motivate the use of CEGs, an asymmetric generalisation.

2.1 Graph Theory

Here we will provide a review of the concepts of graph theory that are fundamental to understand PGMs and therefore underpin the work presented in this thesis.

Definition 2 (Graph) *A graph G is a set of vertices (nodes) $V(G)$ and edges $E(G) \in V(G) \times V(G)$ between the vertices. If the $V(G)$ is finite then G is a finite graph, otherwise it is said to be infinite.*

Definition 3 (Subgraph) *A graph G' is a subgraph of the graph G if $V(G') \subseteq V(G)$ and $E(G') \subseteq E(G)$.*

Definition 4 (Induced subgraph) *An induced subgraph $G' = (V', E')$ induced by vertices $\{v_1, v_2, \ldots, v_n\} \in V(G)$ is a subgraph of $G = (V, E)$ with vertex set $V' = \{v_1, v_2, \ldots, v_n\}$ and edge set $E' = \{(v_i, v_j) | v_i, v_j \in V'\}$.*

Definition 5 (Directed and Undirected Graphs) *A graph G is:*

- *directed if each edge in $E(G)$ has a direction, represented by an arrow showing the direction between vertices.*
- *undirected if all edges in $E(G)$ do not have a direction.*

Definition 6 (Walk, Path and Cycle) *A walk is a series of vertices* $\{v_1, v_2, \ldots, v_n\}$ *such that there is an edge between each consecutive pair of vertices* $(v_i, v_{i+1}) \in E(G)$ *for* $i \in 1, \ldots, n-1$. *A walk in which each vertex appears at most once is called a path. A walk in which the first and last vertex are the same is called a cycle* $v_1 = v_n$.

Definition 7 (Parent and Child) *Given a directed graph* $G = (V, E)$ *a vertex* v *is a parent (or child) of a vertex* $v' \in V$ *if* $(v, v') \in E$ *(or* $(v', v) \in E$*).*

Definition 8 (Directed Acyclic Graph (DAG)) *A DAG is a directed graph with no cycles.*

Definition 9 (Connected graph) *A graph* G *is connected if there exists a path between every pair of vertices.*

Definition 10 (Tree) *A tree* $\mathcal{T} = (V, E)$ *is a connected directed graph with no cycles. It has one vertex called the root vertex* v_0 *with no parents with all other vertices with exactly one parent.*

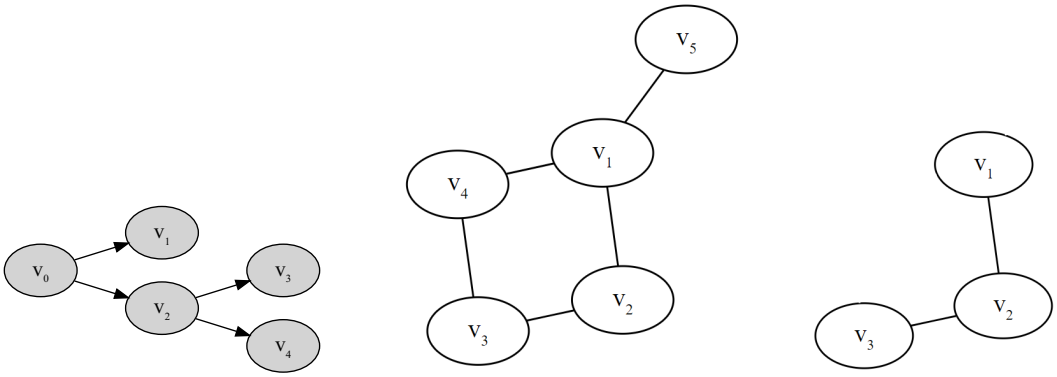


Figure 2.1: Examples of: a tree, a connected graph G and a subgraph of G induced by v_1, v_2, v_3.

Definition 11 (Leaf) *In a tree* $\mathcal{T}$ *a leaf vertex is a vertex with no children.*

2.2 Probabilistic Graphical models (PGMs)

PGMs are a combination of a statistical models and graphical representations of independence relationships [Lauritzen, 1996]. As stated in Section 1.1, their graphical nature makes them interpretable by statisticians, domain experts and stakeholders in decision-making processes, which make them a useful communication tool

in understanding the relationships between variables. Graphs have been used by statisticians in this way for over a century (see e.g. Wright [1921]) and have been developed into a plethora of different representations including influence diagrams [Howard and Matheson, 1981], BNs [Pearl, 1988] and CEGs [Smith and Anderson, 2008]. PGMs provide a strategy to perform effective inference, supporting the creation of a compact representation of probability distributions and providing an intuitive method for expressing assumptions about processes [Pearl, 2009].

2.2.1 Conditional Independence

In PGMs, the graphical structure is often used to represent conditional independence statements. This is a relationship between two variables given the knowledge of a third variable. Variables X and Y are conditionally independent of a third variable Z, denoted:

$$X \perp\!\!\!\perp Y|Z.$$

This means that the probability mass (or density) function, $p(X)$, satisfies the following relation:

$$p(X|Y,Z) = p(X|Z).$$

Here, knowledge about Y does not provide any more information about X if Z is known.

Another type of conditional dependence is that of the context-specific independence; using the same notation as before X is independent of Y given Z is a certain value z:

$$X \perp\!\!\!\perp Y|(Z = z).$$

PGMs represent various factorisations of a probability distribution and, hence, conditional independencies. The graphical structure representing the factorisations can be elicited from expert judgement, but it is commonly learned using structure-learning algorithms.

2.2.2 Bayesian Networks

BNs are a very popular – and well-developed –class of probabilistic graphical model. They were initially developed from Influence Diagrams [Howard and Matheson, 1981; Shachter, 1988]– a graph where decision nodes and utility nodes represent decisions and associated utilities– and later DAGs [Lauritzen, 1996]– graphical models which represent the dependencies between variables. Pearl [1988] first developed BNs in a

bid to create a mathematical framework to represent causality; further developments in technology and statistical research, such as the development of HUGIN (Handling Uncertainty In General Inference Network) [Andersen et al., 1989], have deepened and extended BNs flexibility and adaptability to domain areas. In a BN, each vertex represents a variable with an edge between the variables representing a dependence.

This simple representation is a powerful tool that allows for probabilistic reasoning and inference by using Bayes' theorem to calculate the posterior probability of a variable given evidence.

Example 12 (Anticipating congestion- A BN) *Suppose we want to model the likelihood of traffic congestion (X_C, 'Heavy', 'Light') on a particular road based on the time of day (X_T, 'Day', 'Night') and the weather (X_W, 'Dry', 'Rainy'). Suppose we have elicited that the weather is independent of the time of the day and that both the time of day and weather have an impact on the traffic congestion. These relationships are shown in Figure 2.2.*

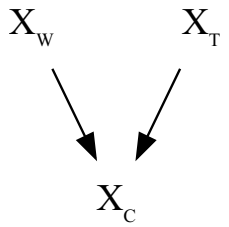


Figure 2.2: Bayesian Network of Traffic Congestion example corresponding to the chance of congestion (X_C) depending on weather (X_W) and time of day (X_T).

Here the probability distribution of each vertex depends on the outcomes of its parents vertex, with the specific values given in probability tables. For example, we may expect more traffic congestion during the day and during rainer weather. This BN can then be used to predict traffic congestion for the weather and time of day which could be used to optimise travel routes.

Whilst Figure 2.2 shows that a relationship might exist between X_C and each of the other variables, we are not able to express more complex statements. We cannot, for example, identify how combinations of the variables X_W and X_T affect X_C from the graph's topology.

2.2.3 Chain Event Graphs

An alternative class of PGMs which could more effectively depict the complex relationships at work here is a CEG, first introduced in Smith and Anderson [2008],

as CEGs are a generalisation of discrete BNs. The development of CEGs was inspired in part by concepts found in Probability Decision Graphs [Jaeger, 2004], BNs [Jensen and Nielsen, 2007] and trees used in decision analysis [Raiffa, 1968].

We provide a brief non-technical review of CEGs; for technical details, see Section 3. CEGs are based on event trees [Shafer, 1996]. Event trees can be embellished with colours, based on their probability distributions, to obtain a CEG [Smith and Anderson, 2008; Collazo et al., 2018]. CEGs are transformations of event trees: they are able to describe the evolution of a process through an unfolding of a sequence of events and give a natural way of talking about events, making them easy for elicitation [Shafer, 1996]. Each (non-leaf) vertex in the tree represents a state an individual may be in and its outgoing edges represent the possible events that follow. Non-leaf vertices are coloured the same if the distribution over their outgoing edges are the same. See Example 13, below.

Example 13 (Anticipating congestion- An Event Tree and CEG) *Continuing Example 12, the event tree in Figure 2.3 shows the variety of ways this process could unfold. This can then be developed, through analysis of the probability distributions over each vertex's edges, into the CEG in Figure 2.4.*

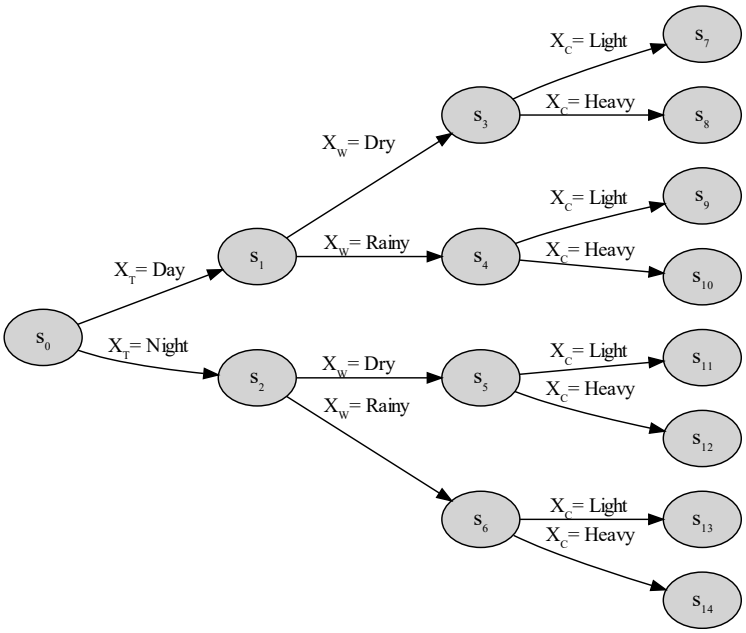


Figure 2.3: Event Tree of Traffic Congestion example corresponding to the chance of congestion (X_C) depending on weather (X_W) and time of day (X_T).

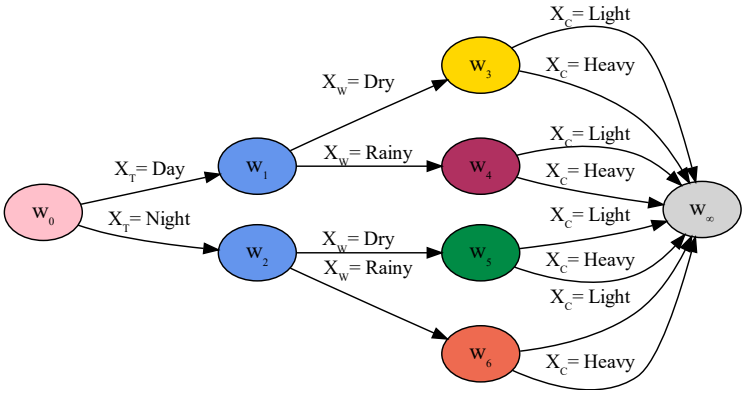


Figure 2.4: A potential CEG of the Traffic Congestion example related to the Event Tree in Figure 2.3

The CEG in Figure 2.4 represents the same dependence structure as that in the BN in Figure 2.2. Note that the topology of the CEG allows us to identify how traffic congestion is impacted by both weather and time of day. The use of colour represents the independence statement that weather is independent of time of day, as both vertices are coloured blue on the graph.

In addition to identifying the independence relationships between different variables, CEGs can also represent context-specific independence statements, such as those in Example 14.

Example 14 (Anticipating Congestion- An alternative CEG) *We continue using Example 13. Whilst Figure 2.4 shows one potential CEG from the Event Tree in Figure 2.3, Figure 2.5 shows an alternative. In this Figure, we see the following independence statements represented:*

- *Blue- Weather is independent of time of day.*
- *Green- Congestion is independent of weather, given that it is night time.*

Representing this context-specific independence statement would not be possible when using a BN.

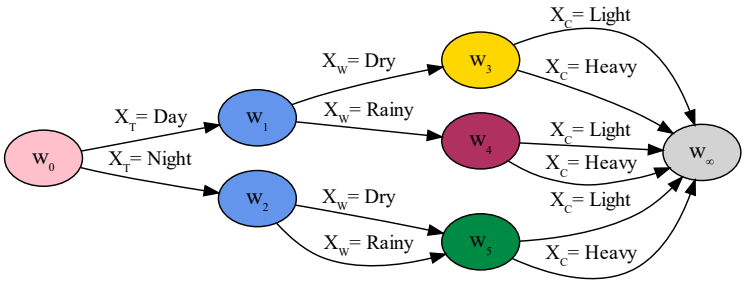


Figure 2.5: An alternative CEG of the Traffic Congestion example related to the Event Tree in Figure 2.3

CEGs can also represent processes which have an asymmetric unfolding of events. Until recently, the vast majority of research has focused exclusively on CEGs which deal with symmetric event spaces– known as *stratified* CEGs. However, recent work has begun to develop an understanding of the unique complexities presented by asymmetric event spaces and their associated *non-stratified* CEGs [Shenvi, 2021; Hughes et al., 2022; Strong et al., 2022; Strong and Smith, 2022b]. Below, we extend our ongoing example to demonstrate an asymmetric event space.

Example 15 (Anticipating Congestion- Asymmetric CEG) *Suppose we add a further variable to the Event Tree in Figure 2.3- X_R, denoting whether a road was closed ('Yes', 'No'). In the case of road closure, it is not logical to consider whether there is congestion on the road as it cannot be used for route planning. The addition of this variable makes the event space asymmetric, by adding a structural missing value, resulting in a non-stratified CEG, as shown in Figure 2.6.*

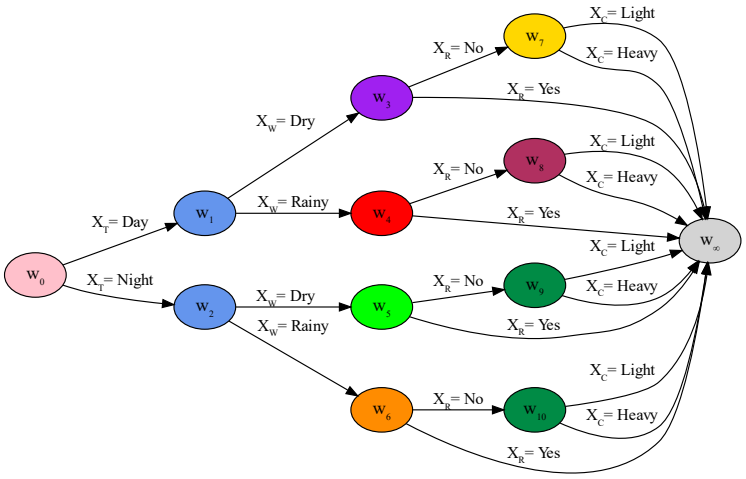


Figure 2.6: A non-stratified CEG of the Traffic Congestion example, with an additional variable, X_R, to denote whether the road was closed

The independence relationships represented in this CEG are as follows:

- *Blue- The weather is independent of time of day.*
- *Green- Congestion is independent of weather, provided the road is open and it is night time.*

Naturally representing this asymmetric unfolding of events would not be possible in a BN– as X_C *has no value when* $X_R = Yes$.

Chapter 3

Chain Event Graphs (CEGs)

Building on the motivation of CEGs in the previous chapter, this chapter provides a review of the topic. This review covers key aspects such as definitions and notation of staged tree and CEGs, including non-stratified CEGs in Section 3.2. Additionally, conjugate learning is discussed in Section 3.3, followed by model selection in Section 3.4. Existing extensions and variants are explored in Section 3.5. In Section 3.6, `cegpy`, a newly-developed Python package, is presented demonstrating its benefits and utility Walley et al. [2022]. Throughout this chapter, we use an example to illustrate the key concepts.

3.1 Introduction

CEGs are a class of interpretable graphical models that can represent asymmetric processes on discrete data. Anderson and Smith [2005] is the source of the first publication on CEGs. As a special case, CEGs include discrete BNs. CEGs generalise finite discrete BNs in two ways: firstly, they can represent more complex independence statements such as context-specific independence statements; secondly, they can represent events that unfold in an asymmetric way. These generalisations occur naturally in many domains. Therefore, due to these additional characteristics, CEGs are extremely versatile and have been applied in a variety of fields, such as policing [Bunnin and Smith, 2021], education [Freeman and Smith, 2011b], migration [Strong et al., 2022], public health [Shenvi et al., 2018] and systems reliability [Yu and Smith, 2021].

Example 16 (Medical Decision Making: Dataset) *To illustrate the concepts in this chapter, we provide an ongoing example using a medical decision-making dataset. This example uses part of the data presented in Trueblood et al. [2018],*

whose research investigates the differences in performance in a medical decision-making task. Experienced pathologists, inexperienced pathologists and novices (undergraduate students) identified whether images of bloods cells were cancerous or not; cancerous and non-cancerous cells are referred to as 'blast' and 'non-blast' cells respectively. Each test subject was given training on identifying blast cells prior to the experiment. Each image of the blood cells was also classified by a separate group of expert pathologists into images that were easy and hard to identify.

For the purposes of our example, we are interested in four variables:

- X_I*: Whether the image was of a cancerous cell or not ("Blast", "Non-blast")*
- X_E*: Experience level of test subject ("Experienced", "Inexperienced", "Novice")*
- X_D*: The difficulty in determining the classification of the cell ("Easy", "Hard").*
- X_R*: Test subject's response ("Blast", "Non-blast")*

3.2 Definitions and Notation

Let $\mathcal{T}$ be a finite event tree, a directed rooted tree, with vertex set $V(\mathcal{T})$ and directed edge set $E(\mathcal{T})$. We define $L(\mathcal{T})$ as the set of leaves in $\mathcal{T}$ and define its complement in $\mathcal{T}$, $S(\mathcal{T}) = V(\mathcal{T}) \setminus L(\mathcal{T})$, as the set of *situations*, non-leaf nodes. Each edge $e \in E(\mathcal{T})$ is an ordered triple (v, v', l) consisting of the vertices the edge originates v and terminates v' and an edge label l. The children of vertex v, denoted $ch(v)$, are the vertices v' for which there exists an l such that $(v, v', l) \in E(\mathcal{T})$. The *floret* of a situation s, $F(s)$, is the subgraph of $\mathcal{T}$ induced by s and its children. Let $\Lambda(\mathcal{T})$ be the set of all root-to-leaf paths, sequences of edges from the root vertex to the leaves along the directed edges, in $\mathcal{T}$. For a path $\lambda \in \Lambda(\mathcal{T})$, let $E(\lambda)$ be the edge set of that path. The root to leaf paths of $\mathcal{T}$ form the atoms of the event space and label all the different possible unfoldings of the process.

Example 17 (Medical Decision Making: Event Tree) *Suppose we decide to use the total variable ordering $X_I < X_E < X_D < X_R$. The event tree representing this process is given in Figure 3.1. To demonstrate some of the notation above, situation $s_5 \in S(\mathcal{T})$ has emanating edges $e_{5,13} = (s_5, s_{13}, Easy)$ and $e_{5,14} = (s_5, s_{14}, Hard)$. The floret of this situation is given by $F(s_5)$, which has vertex set $\{s_5, s_{13}, s_{14}\}$ and edge set $e_{5,13}, e_{5,14}$.*

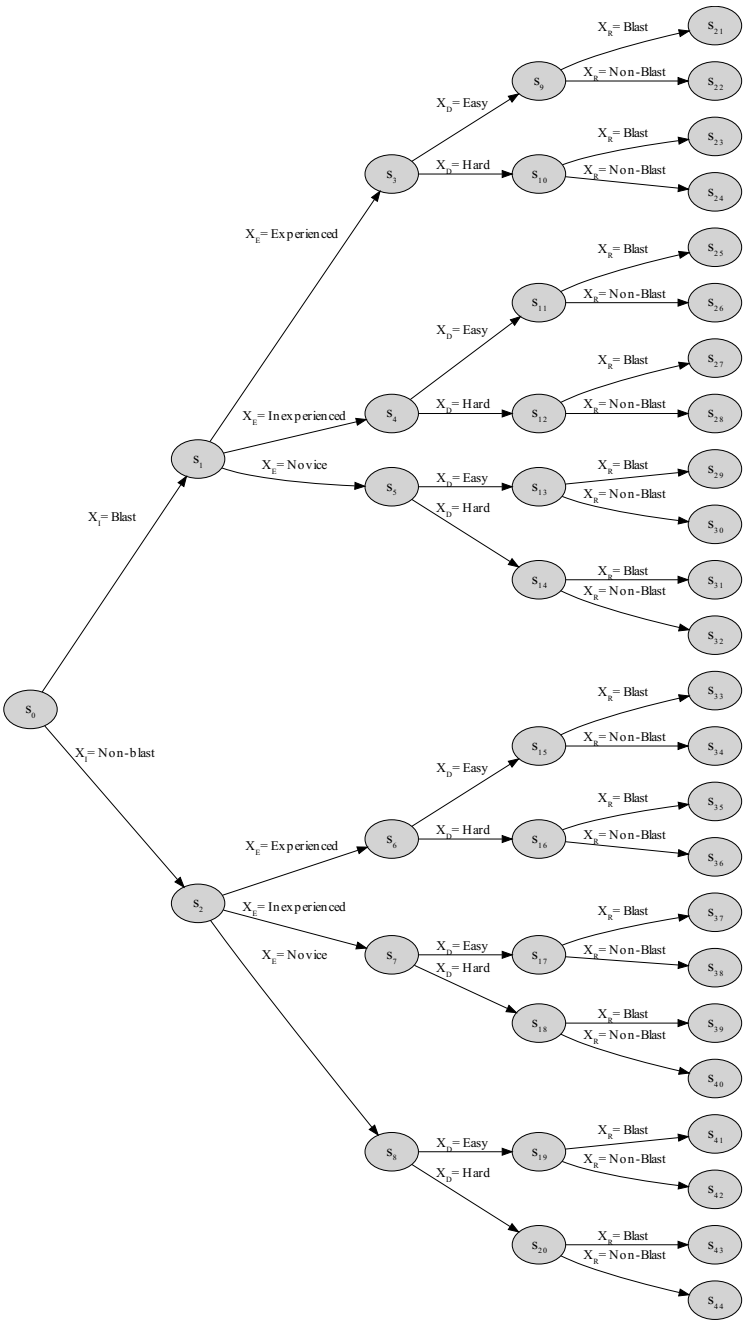


Figure 3.1: Event tree of the medical decision making example.

We define $\Phi_{\mathcal{T}} = \{\boldsymbol{\theta_v} | v \in S(\mathcal{T})\}$ where $\boldsymbol{\theta}_v = (\theta(e) | e = (v, v', l) \in E(\mathcal{T}), v' \in ch(v))$ are the conditional transition probability parameters for each situation, with all conditional transition probabilities being strictly positive that sum to unity over each situation.

The staging of an event tree is a crucial part of modelling with CEGs as

it provides a representation of the dependence structure that exists as part of the process being modelled. This further leads to the compact representation possible in a CEG.

Definition 18 (Stage) *Two situations v and v' in an event tree $\mathcal{T}$ are defined to be in the same stage when $\{\boldsymbol{\theta_v}\} = \{\boldsymbol{\theta_{v'}}\}$. It is also required that the edge labels are the same: that is for $\theta(e) = \theta(e')$ that $e = (v, ., l)$ and $e' = (v', ., l)$ for edges e and e' emanating from v and v' respectively.*

When two situations are in the same stage, it means that their next steps on the root-to-leaf path are equivalent. Therefore, a stage is the set of situations with equivalent conditional transition probability vectors with corresponding edge labels. As discussed in Shenvi [2021], the latter part of this can be relaxed when the edge labels are not fixed i.e. a different recording of a process has the same meaning.

Example 19 (Medical Decision Making: Edge Labels) *Considering again the medical example, suppose we are interested in what the image was of (X_I) and the subject's response (X_R). Both of these variables are recorded as being "Blast" or "Non-blast". This event tree is shown in Figure 3.2. If we instead consider recording the second variable, X_R, as "Was the cell type correctly identified?" with responses ("Correct", "Incorrect"). This would give the event tree shown in Figure 3.3. Stagings on these event trees have different meanings. For example, if in both trees s_1 and s_2 were in the same stage:*

- *In the first tree, this would mean that giving the response "Blast" (or "Non-blast") was independent of the image seen.*
- *For the second tree, this staging has the meaning that the ability to give the correct response is independent of the cell type seen.*

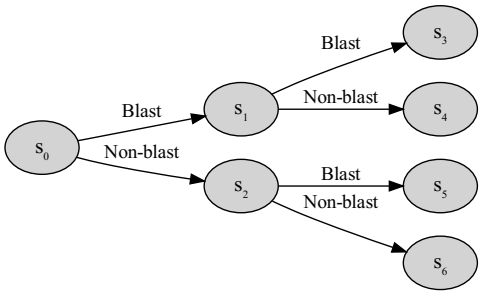


Figure 3.2: An event tree with both variables recorded with "Blast" or "Non-blast".

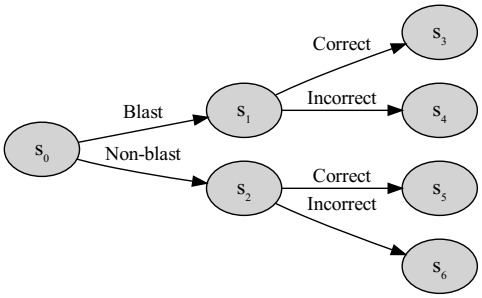


Figure 3.3: An event tree where the first variable is recorded as "Blast" or "Non-blast" and the second is recorded as "Yes" or "No".

The stages, $u \in \mathbb{U}$, are sets of situations which form a partition over the set of all situations. The staging of an event tree is represented through assigning a unique colour to each stage.

Definition 20 (Staged Tree) *We define a staged tree model as $\mathcal{S}$ of an event tree $\mathcal{T}$ where situations are coloured based on the stage they are in with $\Phi_{\mathcal{S}} = \Phi_{\mathcal{T}}$.*

When all situations in a staged tree are in a different stage, the staged tree is called *saturated*.

Example 21 (Medical Decision Making: Staged Tree) *Here, we detail how we can represent independence statements through stage structure. Suppose we knew the following relations between the variables:*

$$X_E \perp\!\!\!\perp X_I$$

$$X_D \perp\!\!\!\perp \{X_E, X_I\}$$

$$X_R \perp\!\!\!\perp X_D | X_I = \text{“Blast”}$$

$$X_R \perp\!\!\!\perp \{X_E, X_D\} | \{X_I = \text{“Non-blast”}, X_E \neq \text{“Novice”}\}$$

These relationships, in plain English, are:

1. *The experience level of the test subject is independent of whether the image was of a blast or non-blast cell.*
2. *The difficulty of determining the classification of the cell is independent of whether the image was of a blast or non-blast cell and the experience level of the test subject.*
3. *The test subject's response is independent of the difficulty of classification, given that the image was of a blast cell.*
4. *The test subject's response is independent of the test subject's level of experience and how difficult classification was, given that the image was of a non-blast cell and the test subject was not a novice.*

This information is recorded in the stage structure of the staged tree in Figure 3.4. The stagings shown in this tree that are not singleton situations are:

$$\{\{s_1, s_2\}, \{s_3, s_4, s_5, s_6, s_7, s_8\}, \{s_9, s_{10}\}, \{s_{11}, s_{12}\}, \{s_{13}, s_{14}\}, \{s_{15}, s_{16}, s_{17}, s_{18}\}\}$$

This example demonstrates how context-specific independence statements can be represented by a staged tree. Stages that consist of a single situation– trivial stages– are often left uncoloured to provide a clearer representation.

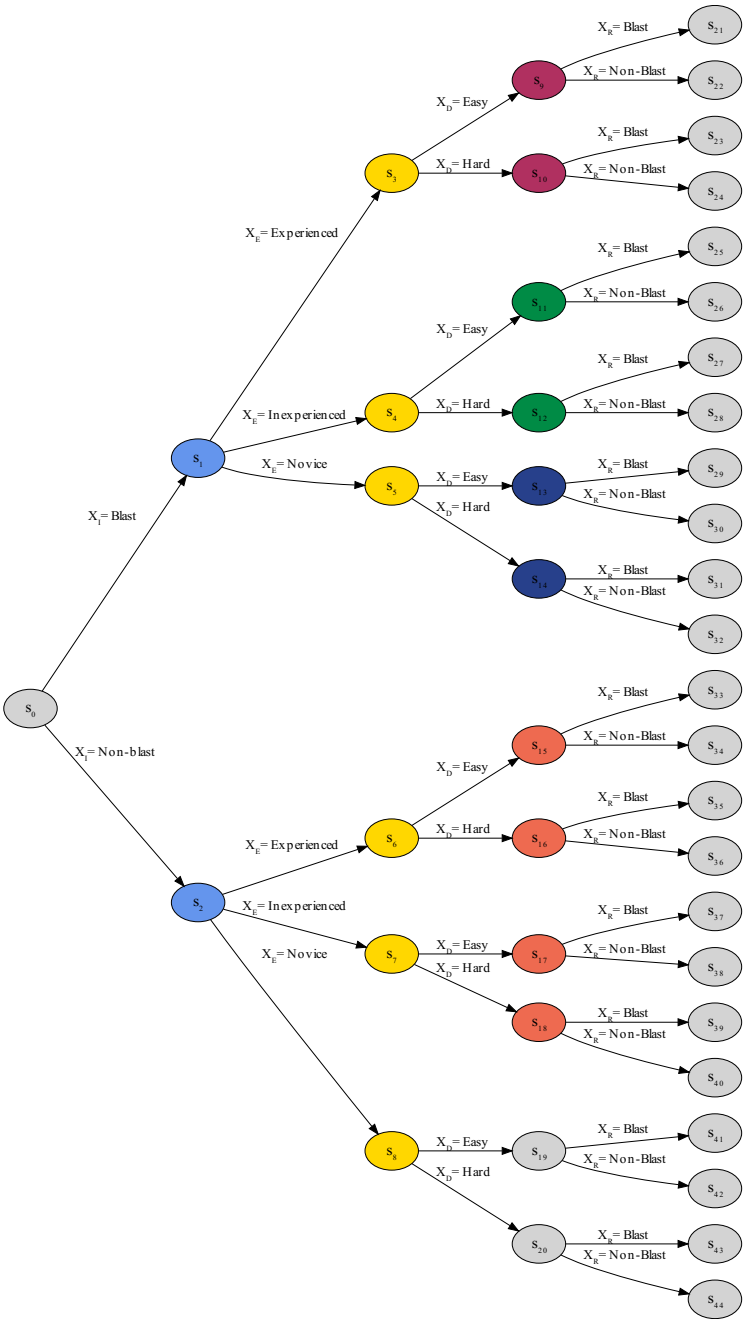


Figure 3.4: Staged tree of the medical decision making example.

In order to obtain a CEG from a staged tree, we need to define position.

Definition 22 (Position) *In a staged tree $\mathcal{S}$, two situations v and v' are said to be in the same position if the sub-trees rooted from v and v' ($\mathcal{S}_v$ and $\mathcal{S}_{v'}$) have the same conditional transition probabilities, $\Phi_{\mathcal{S}_v} = \Phi_{\mathcal{S}_{v'}}$.*

Therefore, situations are in the same stage when the outcomes of their next event have the same probability distribution and they are in the same position if the outcomes of all future events have the same probability distribution. Similarly to stages, the set of position, $w \in \mathbb{W}$, also partitions the set of situations. However, this will be a finer partition.

Example 23 (Medical Decision Making: Position) *Considering the staging given in Example 21, the non-trivial positions are given by the sets:*

$$\{s_6, s_7\}, \{s_9, s_{10}\}, \{s_{11}, s_{12}\}, \{s_{13}, s_{14}\}, \{s_{15}, s_{16}, s_{17}, s_{18}\}$$

Using the concepts of stage and position, we can transform a staged tree into a CEG, as defined in Definition 24 and illustrated in Example 3.5.

Definition 24 (CEG) *A CEG, $\mathcal{C} = (V(\mathcal{C}), E(\mathcal{C}))$, is defined by the triple $(\mathcal{S}, \mathbb{W}, \Phi_{\mathcal{S}})$ with the following properties:*

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

Put simply, a CEG is a staged tree for when any two nodes that have the same distribution over all future events– i.e. they are in the same position– are merged together. This can lead to multiple edges going into a single vertex.

Example 25 (Medical Decision Making: CEG) *The CEG of the medical decision example is given in Figure 3.5.*

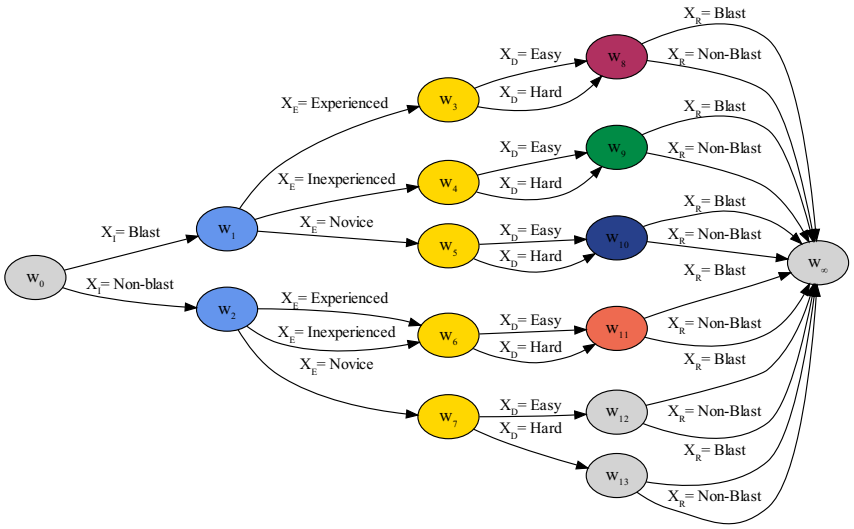


Figure 3.5: CEG of the medical decision making example.

It is important to note that every staged tree can be represented as a CEG as there is a bijective mapping between their representations [Shenvi and Smith, 2020a]. Therefore, they are just different graphical representations of the same model. This naturally raises the question: as there exists a bijective transformation between each staged tree and a CEG, why are we interested in CEGs?

The answer is that a CEG provides a more compact representation as it contains fewer vertices and edges, which is extremely helpful as graphical representations of models based on trees quickly become huge. This makes it easier to read context-specific conditional independencies from the topology of the CEG. The CEG also allows for fast propagation algorithms [Thwaites et al., 2008].

3.2.1 Non-stratified event trees

The class of stratified CEGs has been the focus of most work on CEGs [Shenvi, 2021]. However, the wider set of non-stratified CEGs is able to represent structural asymmetries. Given a vector $\mathcal{X} = \{X_1, X_2, \ldots, X_n\}$ of variables, we define $\mathcal{X}^k = \{X_1, X_2, \ldots, X_k\}$ for $1 \leq k \leq n$ and the state space of variable X_i as $\mathbb{X}_i$.

Definition 26 ($\mathcal{X}$-compatible) *An event tree $\mathcal{T}$ is $\mathcal{X}$-compatible if its vertex set $V(\mathcal{T})$ consists of a root node v_0 together with a vertex $v(x^k)$ for each $x^k = (x_1, x_2, ..., x_k)$ where $x_i \in \mathbb{X}_i$ and $1 \leq k \leq n$.*

This means that there is a vertex in the tree for every possible ordered combination of the variables.

Definition 27 ($\mathcal{X}$-stratified) *A staged tree is said to be an $\mathcal{X}$ stratified staged tree when its underlying event tree is $\mathcal{X}$-compatible.*

In this thesis, we define a staged tree as stratified (as in Cowell and Smith [2014] and Shenvi [2021]) if it is $\mathcal{X}$-stratified for some $\mathcal{X}$.

We therefore define non-stratified CEGs as the CEGs transformed from non-stratified event trees. These naturally arise in real-world systems. We give an example of this below. The methods describe in this thesis apply generally to both stratified and non-stratified CEGs.

Note that an alternative definition of stratified staged trees has been used [Collazo et al., 2018], in which a tree is stratified if all situations in the same stage are the same distance from the root. These definitions are not equivalent as a staged tree with structural zeros would be classified as stratified in this alternative definition.

Example 28 (Medical Decision-Making: Non-stratified Extension) *Suppose that in the medical decision-making task we wanted to ask about the process of how someone identified a cell. This could be done by asking the participant a series of questions on what they observed before asking them to classify the cell. Suppose this was done in the following way: the participants were shown an image of a cell then first asked if the cell has a nucleus, if it did whether it was small or large, and then finally all participants were asked if the cell was blast or not.*

This process can be seen represented in Figure 3.6. This gives an example of a non-stratified tree. This is an example of a tree that includes a structural missing value, a value which is missing which has no underlying meaningful value.

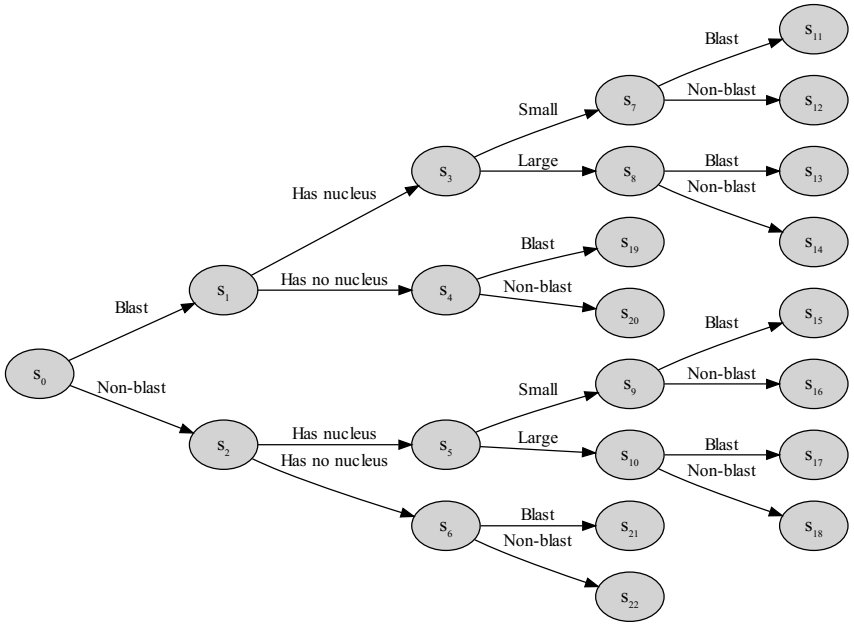


Figure 3.6: Event tree of the medical decision making non-stratified extension example.

Suppose further that it is known by all participants that a "Blast" cell does have a nucleus; therefore, if the participant says the cell does not have a nucleus then they must think the cell is not a blast cell. This introduces a structural zero, a path in which observing a count is logically impossible. These logically impossible paths are removed from the tree, giving the event tree seen in Figure 3.7.

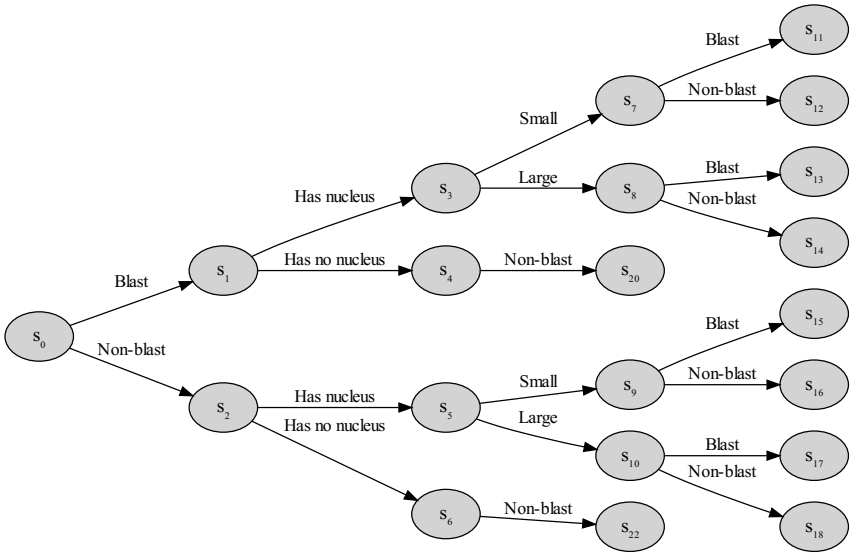


Figure 3.7: Event tree of the medical decision making non-stratified extension example with structural zeros.

3.3 Conjugate Learning

In the previous section, we have demonstrated how a CEG can represent structural information in the form of dependencies between events. Here, we show how to perform a conjugate updating of the parameters in a CEG as detailed in Freeman and Smith [2011a] and Collazo et al. [2018].

In this updating, posterior probabilities for the CEG can be obtained through a Dirichlet-Multinomial conjugate updating over each situation's edges. Performing a conjugate analysis is attractive as it leads to interpretable hyperparameters and a closed form updating of the posterior. This methodology closely follows the framework for conjugate learning developed for discrete BNs [Heckerman et al., 1995].

Suppose we have a CEG $\mathcal{C}$ with K stages labelled $\{u_1, \ldots, u_K\} \in \mathbb{U}$, with stage i having k_i outgoing edges. The conditional transitional probability of each stage is given by $\boldsymbol{\theta}_i = \{\theta_{i1}, \theta_{i2}, \ldots, \theta_{ik_i}\}$. Here θ_{ij} is the probability of going along the jth edge for a situation $s \in u_i$ for $j \in \{1, 2, \ldots, k_i\}$. We denote a random sample with no missing data as $\boldsymbol{y} = \{\boldsymbol{y_1}, \boldsymbol{y_2}, \ldots, \boldsymbol{y_K}\}$ with each $\boldsymbol{y}_i = \{y_{i1}, y_{i2}, \ldots, y_{ik_i}\}$ where y_{ij} is the number of individuals that start in situation $s \in u_i$ and transfer along its jth edge.

Provided that the sampling experiment was properly randomised, θ_i will have a multinomial distribution. Hence, $\theta_{ij} \geq 0$ and $\sum_{j=1}^{k_i} \theta_{ij} = 1$. As demonstrated in Example 28, we will assume all structural zero paths have been removed from the

tree and therefore $\theta_{ij} > 0$. We further take the local and global independence assumptions, that the transition probabilities are mutually independent *a priori*.

Therefore, the likelihood that CEG $\mathcal{C}$ can be written as a product of likelihood of florets, which in turn can be written as a product of edge probabilities, as follows:

$$\begin{aligned} p(\boldsymbol{y}|\Phi_{\mathcal{C}}, \mathcal{C}) &= \prod_{i=1}^{K} p(y_i|\boldsymbol{\theta}_i, \mathcal{C}) \\ &= \prod_{i=1}^{K} \prod_{j=1}^{k_i} \theta_{ij}^{y_{ij}} \end{aligned} \tag{3.1}$$

where $\Phi_{\mathcal{C}} = \{\boldsymbol{\theta}_i | u_i \in \mathbb{U}\}$. To perform a conjugate analysis, as in for BN modelling, for each $\boldsymbol{\theta}_i$, we set a Dirichlet prior distribution with parameter vector $\boldsymbol{\alpha}_i = (\alpha_{i1}, \alpha_{i2}, \ldots, \alpha_{ik_i})$ where $\alpha_{ij} > 0, j \in \{1, 2, \ldots, k_i\}$:

$$\begin{aligned} p(\Phi_{\mathcal{C}}) &= \prod_{i=1}^{K} p(\boldsymbol{\theta}_i|\mathcal{C}) \\ &= \prod_{i=1}^{K} \frac{\Gamma(\bar{\boldsymbol{\alpha}}_i)}{\prod_{i=1}^{k_i} \Gamma(\alpha_{ij})} \prod_{j=1}^{k_i} \theta_{ij}^{\alpha_{ij}-1}. \end{aligned} \tag{3.2}$$

Here, the Gamma function is denoted by $\Gamma(z) = \int_0^\infty x^{z-1} \exp(-x) dx$ and we use the notation $\bar{\boldsymbol{\alpha}} = \sum_{i=1}^{n} \alpha_i$ for a vector $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_n)$. Using Equations 3.1 and 3.2, the posterior distribution of $\boldsymbol{\theta}_i$ can be obtained as follows:

$$\begin{aligned} p(\boldsymbol{\theta}_i|\boldsymbol{y_i}, \mathcal{C}) &\propto p(\boldsymbol{\theta}_i|\mathcal{C}) \prod_{j=1}^{k_i} p(y_{ij}|\boldsymbol{\theta}_i, \mathcal{C}) \\ &\propto \prod_{j=1}^{k_i} \theta_{ij}^{\alpha_{ij}-1} \theta_{ij}^{y_{ij}} \\ &= \prod_{j=1}^{k_i} \theta_{ij}^{\alpha_{ij}+y_{ij}-1}. \end{aligned} \tag{3.3}$$

This shows that the posterior distribution of $\boldsymbol{\theta}_i$ is also Dirichlet with parameter vector $\boldsymbol{\alpha}_i^* = (\alpha_{i1}^*, \alpha_{i2}^*, \ldots, \alpha_{ik_i}^*)$ where $\alpha_{ij}^* = \alpha_{ij} + y_{ij}$, $i \in \{1, 2, \ldots, K\}$, $j \in \{1, 2, \ldots, k_i\}$. This means that, in a conjugate analysis, the parameters of each stage can be quickly updated independently of each other due to the closed form

representation.

Another advantage of the closed form nature of the conjugate analysis is that the marginal likelihood can also be written as follows:

$$\begin{aligned} p(\boldsymbol{y}|\mathcal{C}) &= \int_{\Phi_{\mathcal{C}}} \prod_{i=1}^{K} \left\{ p(y_i|\boldsymbol{\theta_i}, \mathcal{C}) p(\boldsymbol{\theta}_i|\mathcal{C}) \right\} d\Phi_{\mathcal{C}} \\ &= \int_{\Phi_{\mathcal{C}}} \prod_{i=1}^{K} \left\{ \prod_{j=1}^{k_i} \theta_{ij}^{y_{ij}} \times \frac{\Gamma(\bar{\boldsymbol{\alpha}}_i)}{\prod_{i=1}^{k_i} \Gamma(\alpha_{ij})} \prod_{j=1}^{k_i} \theta_{ij}^{\alpha_{ij}-1} \right\} d\Phi_{\mathcal{C}} \\ &= \prod_{i=1}^{K} \left\{ \frac{\Gamma(\bar{\boldsymbol{\alpha}}_i)}{\Gamma(\bar{\boldsymbol{\alpha}}_i^*)} \prod_{j=1}^{k_i} \frac{\Gamma(\alpha_{ij}^*)}{\Gamma(\alpha_{ij})} \right\}. \end{aligned} \tag{3.4}$$

This is useful in being able to perfom model selection. To assess the performance of a model, we are interested in the following equation:

$$p(\mathcal{C}, \boldsymbol{y}) = p(\mathcal{C}) p(\boldsymbol{y}|\mathcal{C}). \tag{3.5}$$

Here, $p(\mathcal{C})$ is the prior of CEG $\mathcal{C}$. This is the general form of the Bayesian Dirichlet (BD) score, where different choices of the Dirichlet parameters give different versions of the BD score. For BNs, there are several ways of setting these priors, each with attractive properties [Heckerman et al., 1995; Scutari, 2018]. For details of how the priors are set, see Section 3.3.1.

This score can alternatively be written as:

$$\log p(\mathcal{C}, \boldsymbol{y}) = \log p(\mathcal{C}) + \log p(\boldsymbol{y}|\mathcal{C}) \tag{3.6}$$

Typically, a uniform prior is set over the set of all possible models, meaning that all stages of CEGs are equally likely. This simplifies the model selection to only depending on the marginal likelihood given in Equation (3.4); for CEG $\mathcal{C}$, we will refer to this as $Q(\mathcal{C})$. The log marginal likelihood can be written as:

$$Q(\mathcal{C}) = \sum_{i=1}^{K} \left\{ g(\bar{\boldsymbol{\alpha}}_i) - g(\bar{\boldsymbol{\alpha}}_i^*) + \sum_{j=1}^{k_i} g(\boldsymbol{\alpha}_{ij}^*) - g(\boldsymbol{\alpha}_{ij}) \right\} \tag{3.7}$$

and $g(x) = \log \Gamma(x)$. This simple additive form will be important for comparing competing models, which we will cover in Section 3.4.

3.3.1 Prior Setting

As with all Bayesian modelling, it is important that the priors are set up logically. Suppose we have an elicited CEG $\mathcal{C}$; we should consider how to set the $\boldsymbol{\alpha}$ Dirichlet prior, as used in Section 3.3.

There are two main ways of setting hyperparameters for the Dirichlet priors within a CEG. The first approach uses expert elicitation and involves setting the prior means and the imaginary sample size for each stage; the second involves propagating an imaginary sample size uniformly across the edges of the CEG graph.

In the first case, we set the hyperparameters of each stage in the CEG independently. For any given stage u_i, the Dirichlet prior is $Dir(\boldsymbol{\alpha}_i)$ with $\boldsymbol{\alpha}_i = (\alpha_{i1}, \alpha_{i2}, \ldots, \alpha_{ik_i})$, where k_i is the number of outgoing edges from stage u_i, and α_{ij} is often interpreted as pseudo-counts or imaginary sample size of stage i along its jth emanating edge.

This prior can be calculated by decomposing it into the prior means and total effective sample size for that stage, μ_{ij} and $\bar{\alpha}_i$ respectively, with $\alpha_{ij} = \bar{\alpha}_i \mu_{ij}$. This decomposition can be interpreted as the expected probability of transferring along each edge multiplied by $\bar{\alpha}_i$, where $\bar{\alpha}_i$ is a measure of strength in this belief. μ_{ij} and $\bar{\alpha}_i$ can both be elicited from domain experts. This can unfortunately be a huge challenge as it can be a very time consuming process and is not always possible especially if the domains relating to the priors are varied, needing different experts.

Alternatively, we can make use of a mass conservation property in order to set the Dirichlet priors [Collazo et al., 2018; Hughes et al., 2022]. This simply states that the imaginary sample size of the edge into any situation s is equal to the sum of the effective sample sizes of the edges emanating from s. This means that the sum of the effective sample size into the leaves is the same as the total number at the root of the tree.

The hyperparameters can be set using this approach by simply choosing an effective sample size at the root, $\bar{\boldsymbol{\alpha}}_0$, which is spread across the graph, in a way elicited from domain experts. The mass conservation property means that the prior on the stages closer to the root vertex will have a larger effective sample size than those further away.

When expert elicitation is not available, this approach can be used to define a default way of setting hyperparameters. An effective sample size still needs to be chosen; this can then be propagated across the edges such that the sample size at the leaves is uniform, giving the Bayesian Dirichlet equivalent path uniform prior (BDepu) [Hughes et al., 2022]. Alternatively, we can split the effective sample size uniformly over every edge encountered from the root to the leaves, giving the

Bayesian Dirichlet equivalent uniform prior ((BDeu) [Cowell and Smith, 2014]. For a stratified CEG, these two methods of setting the prior are equivalent [Hughes et al., 2022].

By setting a weakly informative prior as the default prior, we make sure that it does not have large influence over the model. A typical heuristic choice of the effective sample size of the prior is to set it as the maximum number of outgoing edges from any vertex in the graph [Neapolitan, 2003].

Whilst these provide an easier way of setting priors, like any heuristic there are situations when this is not suitable and care should be taken to make sure the priors are appropriately set. For example, setting a uniform prior over all the leaves will not be suitable if there are any structural zeros in the tree representing the process.

3.4 Model Selection

We have detailed how a Bayesian analysis can be performed when the given CEG is known. However, this is not always the case and in these instances we need to be able to determine which model is the best representation of the data-generating process.

As detailed in Section 3.3.1, we know how to set the priors up for an elicited CEG. However, this is infeasible when there is a huge set of models to consider. We can use calibrated priors to mitigate this problem, in which we choose a model– often the saturated model (with no structure)– to set the priors on and use them for all models. If the non-saturated model is chosen to select the priors, the mass conservation property and stage independence can be used to determine what the other priors should be.

Once the priors are calibrated to compare different CEGs, we can compare their performance. There are many different ways of performing model selection for CEGs; the approaches currently explored in the research have predominately been score-based. These functions include the Bayesian information criterion [Schwarz, 1978], Akaike's information criterion [Akaike, 1974] and factorised normalised maximum likelihood [Silander et al., 2010].

In this thesis, we will use *Maximum A Posteriori* (MAP) model selection, although the methods developed throughout this thesis are applicable to many score functions. MAP model selection is used to search for the model we are interested in, $\mathcal{C}$, that maximises the joint probability of the data and the model, as given in Equation (3.5). The MAP model has some attractive properties such as consistency,

$\log BF(\mathcal{C}, \mathcal{C}')$	$BF(\mathcal{C}, \mathcal{C}')$	Evidence against $\mathcal{C}'$
0-1.10	1-3	Not worth more than a bare mention
1.10-3	3-20	Positive
3-5	20-150	Strong
>5	>150	Very strong

Table 3.1: An interpretation of Bayes Factor [Kass and Raftery, 1995].

given enough data and under certain conditions: if the data-generating model is in the set of considered models, it will be chosen. If the data-generating model is not in the set of models, and the same conditions hold, given enough data, the MAP model will be the one with the shortest Kullback-Leibler divergence from the data-generating model [Bernardo and Smith, 2009].

The MAP model selection also allows for flexibility based on how the priors – both over the set of models and for the Dirichlet distributions– are set. When using a uniform prior over the set of models, the MAP model is that which maximises the marginal likelihood.

To quantify the differences between models, we can use the Bayes factor (BF), the fraction of marginal likelihoods for different models. For CEGs, $\mathcal{C}$ and $\mathcal{C}'$ the $\log BF$ is written as:

$$\log BF(\mathcal{C}, \mathcal{C}') = Q(\mathcal{C}) - Q(\mathcal{C}'). \tag{3.8}$$

An interpretation on the strength of different BFs is given in Table 3.1, although these choices are somewhat subjective and are well-known to depend on context Kass and Raftery [1995]. However, they are widely-used and provide a rough guide of the strength of evidence between models.

Before discussing model selection algorithms, it is important to consider what is in the set of models we are considering. In this section, we focus on models all based on a single tree; we discuss different orderings of the tree in Section 3.4.5. In this case, this means model selection is determining the staging of the event tree; model selection for a given tree is determined solely on how the situations are coloured, due to the bijection between staged trees and CEGs.

3.4.1 Hyperstages

Here, we introduce the concept of the hyperstage, this restricts the set of stagings to those that are logically plausible [Collazo, 2017]. For example, two situations with different numbers of outgoing edges cannot be in the same stage. Less obviously,

in any given context, for two edge probabilities to be assumed equal, we often want to associate the meaning of these edges in some way. Therefore when performing model selection, it is important, both from a modelling and a computational point of view, to restrict the search space so that only models that are logical within a given context are traversed and scored. Clearly, the choice of this restricted space can depend strongly on the domain.

Definition 29 (Hyperstage) *A hyperstage* $\boldsymbol{H} = \{H_1, H_2, ..., H_n\}$ *is a collection of sets, hypersets, such that any two situations* $v, v' \in S(\mathcal{T})$ *can only be in the same stage in* $\mathbb{U}$ *if there is a set* $H_i \in \boldsymbol{H}$ *such that* $v, v' \in H_i$.

This hyperstage typically corresponds to a partition of the set of situations where, before any data analysis has taken place, situations within the same set could be plausibly seen as predictively equivalent, were data to support this *a posteriori*. This simplifies model selection as the staging of each set in the hyperstage can be done independently. Therefore, we can perform model selection on each hyperset separately.

We note, for example, in all discrete BNs, we implicitly assume that conditional probabilities can only be hypothesised as being the same when the situations defined by their parents involve the same variables [Collazo et al., 2018]. A standard way to set the hyperstage when there are no structural zeros is to have a hyperset for each variable (or event) of the same type. In a stratified CEG, this corresponds to situations that are the same distance from the root being in the same hyperset. In a non-stratified CEG, the default way of setting the hyperstage is when situations have the same number of outgoing edges and the same edge labels. However, this is not always appropriate when variables share the same edge labels but a different real-world meaning, as shown in the following example.

Example 30 (Medical Decision-Making: Hyperstage setting) *This example uses the event tree in Figure 3.6 to illustrate how hyperstages are set in non-stratified event trees. The default way of setting the hyperstage for this tree would be as follows:*

$$\boldsymbol{H} = \{\{s_0, s_4, s_6, s_7, s_8, s_9, s_{10}\}, \{s_1, s_2\}, \{s_3, s_5\}\}.$$

However, if we return to the original definitions of the variables in Example 16, we notice that the variable represented by s_0 *has a different real-world meaning as it relates to a different variable– the type of image seen)– to the other situations in its hyperset, which represent variable* X_R *(the response given). Therefore, a more*

suitable hyperstage for this dataset would be:

$$\{\{s_0\}, \{s_4, s_6, s_7, s_8, s_9, s_{10}\}, \{s_1, s_2\}, \{s_3, s_5\}\}.$$

In this case, it is not logical to consider the distribution governing the image seen as the same as the response given.

3.4.2 Square-free CEGs

In this thesis, we focus our attention onto a subclass of CEGs, those which are *square-free* [Collazo et al., 2018].

Definition 31 (Square-free) *A CEG $\mathcal{C}$ is square-free if no two situations that lie on the same root-to-leaf path are in the same stage.*

Through setting the hyperstage, a CEG can be guaranteed to be square-free. However, it is important to note that this restriction may not always be suitable, particularly when a root-to-leaf path involves recurrent events. Nevertheless, when this is not the case, this restriction is a natural one. Square-free CEGs are the focus in most of this thesis as many assumptions about the independence of priors and situations may not be applicable outside of this context.

3.4.3 Cardinality

For any given probability tree, each distinct staging will give rise to a different model in our model space.The number of staged trees in the model space, is given by a product of Bell numbers that grow super-exponentially [Silander and Leong, 2013]. Given a stratified CEG, with the default hyperstage and $N > 0$ events and each event having two outcomes, the number of different staged trees, $\#\mathcal{S}$, is given by:

$$\#\mathcal{S} = \prod_{i=1}^{N} B(2^{N-1}) \tag{3.9}$$

where $B(m)$ is the m^{th} Bell number. For example, when $N = 5$, there are approximately 1.3×10^{15} possible models. Therefore, an exhaustive search through this space quickly becomes infeasible as the number of events represented by the event tree increases.

To overcome this issue, there have been various model selection algorithms used to explore the space. These include but are not limited to: Agglomerative Hierarchical Clustering (AHC) [Freeman and Smith, 2011a], dynamic programming

[Cowell and Smith, 2014], K-means [Silander and Leong, 2013] and mixture modelling approaches [Shenvi and Liverani, 2022]. More detail of model selection algorithms for CEGs is given in Section 5.2.

3.4.4 AHC

For now, we will focus our attention on the most popular model selection algorithm, AHC. This is done using *one-nested* CEGs: CEGs which can be obtained from others by merging two stages. Comparing one-nested CEGs simplifies calculating the BF. All of the stages which are the same in both CEGs cancel out, leaving just the stages which are different between the two models left. For CEGs $\mathcal{C}$ and $\mathcal{C}'$, the $\log BF(\mathcal{C}, \mathcal{C}')$, where $u_{i\oplus j}$ is the stage in $\mathcal{C}'$ obtained by combining stages u_i and u_j from $\mathcal{C}$, is given by:

$$\begin{aligned}\log BF(\mathcal{C}, \mathcal{C}') =& g(\bar{\boldsymbol{\alpha}}_{i\oplus j}) - g(\bar{\boldsymbol{\alpha}}_i) - g(\bar{\boldsymbol{\alpha}}_j) - g(\bar{\boldsymbol{\alpha}}^*_{i\oplus j}) + g(\bar{\boldsymbol{\alpha}}^*_i) + g(\bar{\boldsymbol{\alpha}}^*_j) \\ &+ \sum_{l=1}^{k} \left\{ g(\alpha^*_{i\oplus jl}) - g(\alpha^*_{il}) - g(\alpha^*_{jl}) - g(\alpha_{i\oplus jl}) + g(\alpha_{il}) + g(\alpha_{jl}) \right\}.\end{aligned} \tag{3.10}$$

Here, $\alpha_{i\oplus j}$ gives the hyperparameters of the mergings of stages i and j. This enables fast evaluation of the comparison as the log BF of one-nested CEGs can be calculated by only considering situations in which their stagings are different.

AHC is a greedy search algorithm which takes the locally optimal step at each choice. At the start of the process, each situation as a separate stage. Then, AHC calculates the BF compared to all one-nested models and merges stages that lead to the largest improvement [Freeman and Smith, 2011a]. The pseudocode for the AHC algorithm is given in Algorithm 1.

Input : Event tree $\mathcal{T}$, its associated hyperstage $\boldsymbol{H}$, data $\mathbf{y}$ and root equivalent sample size $\bar{\boldsymbol{\alpha}}_0$.

Output: A CEG and its associated staging and log marginal likelihood score

Initialise *data*, y_i for each situation s_i in $\mathcal{T}$ from $\mathbf{y}$.

Initialise *priors*, $\boldsymbol{\alpha}_i$ for each situation s_i in $\mathcal{T}$ from $\bar{\boldsymbol{\alpha}}_0$ through mass conservation.

Initialise a *stage* for each situation s_i in $\mathcal{T}$.

Set *score* as the log marginal likelihood score given in Equation (3.7).

Set *indicator* $\leftarrow 1$.

while *indicator* $\neq 0$ **do**

  if *There is a single stage in every hyperstage* **then**

    indicator $\leftarrow 0$

  end

  for *every pair of stages in stages in the same hyperstage* **do**

    Calculate the $\log BF$ as given in Equation (3.10) comparing the structures of merging the pair to keeping them apart, all other stages being equal.

  end

  if *There exists a calculated logBF* ≥ 0 **then**

    for *pair* u_i *and* u_j *with the largest logBF* **do**

      score $\leftarrow$ *score* $+ \log BF(u_i, u_j)$

      Update *stages* to add stage $u_{i\oplus j}$ and remove stages u_i and u_j.

      Update *data* to add $y_{i\oplus j} = y_i + y_j$ and remove y_i and y_j.

      Update *priors* to add $\boldsymbol{\alpha}_{i\oplus j} = \boldsymbol{\alpha}_i + \boldsymbol{\alpha}_j$ and remove $\boldsymbol{\alpha}_i$ and $\boldsymbol{\alpha}_j$.

    end

  end

  else

    indicator $\leftarrow 0$

  end

end

return *stage, score*

Algorithm 1: AHC algorithm

3.4.5 Variable ordering

We briefly discuss research that has been done when the ordering of events is not known, or with a block ordering. This research focuses on stratified CEGs. For

a stratified CEG with N events, there are $N!$ different orderings of the events. Identifying the most effective ordering quickly becomes intractable for all but the smallest of problems.

Dynamic programming has been used to find the best possible ordering [Silander and Leong, 2013; Cowell and Smith, 2014] using the fact that, given N events, the best ordering of $N-1$ events will be the same as the best ordering for all N variables.

Alternative approaches have been suggested for stratified trees that involve learning a BN first and then extending on that model to learn the context-specific independencies [Collazo and Taranti, 2017; Leonelli and Varando, 2022].

3.5 Extensions and Variants

Several extensions have been developed to the standard CEG. Here, we present a succinct overview of a select few.

3.5.1 Ordinal CEGs

Ordinal CEGs were introduced in Barclay et al. [2014]; they are CEGs on binary variables that vertically align and order the positions of stages so that the probability of outcomes decreases for each variable the further down. This ordering could be used to inform inference and search. However, to date, this is a purely cosmetic change in order to increase the readability of CEGs to assess how probable certain events are.

3.5.2 Dynamic Variants

Dynamic extensions of CEGs exist in two main types. The first is an infinitely large tree that can represent potentially infinite reoccurring processes [Barclay et al., 2015] and can include holding times on their edges which can also be staged [Shenvi and Smith, 2020b]. This describes the evolution of units that can continue indefinitely within a given population. The second extension is a dynamic staging in which the dependence structures within a finite tree is modelled to change over time, as in Freeman and Smith [2011b]. This imposes a dynamic development on the edge probability of the tree and its structure as this applies to otherwise replicating populations over each time interval.

3.6 Software for Chain Event Graphs

In this section, we will discuss the development of an open source software tool called `cegpy`, which is designed to facilitate model selection for both stratified and non-stratified CEGs from data. This section illustrates work from Walley et al. [2022], a paper on which I am a co-author. It details the motivation for this package, a demonstration of some of its functionality and a discussion on future extensions.

3.6.1 Introduction and motivation

Asymmetric processes are prevalent in many real-world situations, yet they remain challenging to model accurately. Despite the proven flexibility offered by CEGs in these circumstances, CEGs are yet to be widely adopted by applied statisticians. This is primarily due to the lack of existing software, particularly for modelling structurally-asymmetric processes.

In contrast, there exist several well-developed and maintained software for modelling with BNs. These include Netica [Norsys Software Corp, 2020], Weka [Eibe et al., 2016], BARD [Nyberg et al., 2022], GeNIe [BayesFusion, LLC, 2022], and HUGIN [HuginExpert, 2022], and coding packages such as `bnlearn` [Scutari, 2010] and `deal` [Bottcher and Dethlefsen, 2018] in R and `BayesPy` [Luttinen, 2016], `GOBNILP` [Cussens, 2020] and `BayeSuites` in Python.

Whilst there are two existing packages that can learn and visualise CEGs from data, representing context-specific independence statements– the R packages `ceg` [Collazo and Taranti, 2017] and `stagedtrees` [Carli et al., 2020]– neither can represent non-stratified CEGs, which are necessary for modelling processes with asymmetric unfoldings of events. `cegpy` fills this gap: it is the first package that can learn and visualise non-stratified CEGs from data and the first CEG package available for `python`. Its novel contribution fills a significant gap in the existing software landscape; the development of `cegpy` means that applied statisticians can now take advantage of the flexibility offered by CEGs to model a wide range of asymmetric processes more effectively.

`cegpy` uses path-based approach where all the data is associated with edges of the event tree and uses events as the building blocks of the model, facilitating the capacity to routinely handle non-stratified CEGs. Contrastingly, `ceg` and `stagedtrees` use a column-based approach, which associates the data to the variables of the model and to their corresponding state spaces. This approach makes it extremely difficult to model non-stratified CEGs.

3.6.2 Functionality

In this section, we illustrate `cegpy`'s key functionalities. We consider again the ongoing example on medical decision-making. Note that illustrations and guidance for the full range of functionalities supported by `cegpy`, including probability propagation, can be found at https://cegpy.readthedocs.io with examples in the binder: https://github.com/peterrhysstrong/cegpy-binder.

The Dataset

Below, we demonstrate how `cegpy` can be used to perform model selection. Note that here we have continued to use the dataset in Example 16, earlier in this chapter, for continuity purposes; for non-stratified examples of `cegpy`'s application, see illustrations in Walley et al. [2022] and research using `cegpy` as a tool for developing non-stratified CEGs in Chapters 4 and 5.

```
from cegpy import StagedTree, ChainEventGraph
import pandas as pd

df = pd.read_excel("medical.xlsx")
print(df.head(5))
```

```
output:
  Classification        Group Difficulty   Response
0          Blast  Experienced       Easy      Blast
1      Non-blast  Experienced       Easy  Non-blast
2      Non-blast  Experienced       Hard      Blast
3      Non-blast  Experienced       Hard  Non-blast
4          Blast  Experienced       Easy      Blast
```

The Event Tree

We initialise an EventTree object using the following code. Figure 3.8 gives an illustration of the code's output.

Since `cegpy` constructs the event tree by creating a dictionary of the paths in the input dataset, there is no need to specify structural zeros as they do not occur in the dataset. Structural missing values are identified in the dataset as NaNs. However, the medical dataset has neither of these features.

```
event_tree = EventTree(df)
event_tree.create_figure()
```

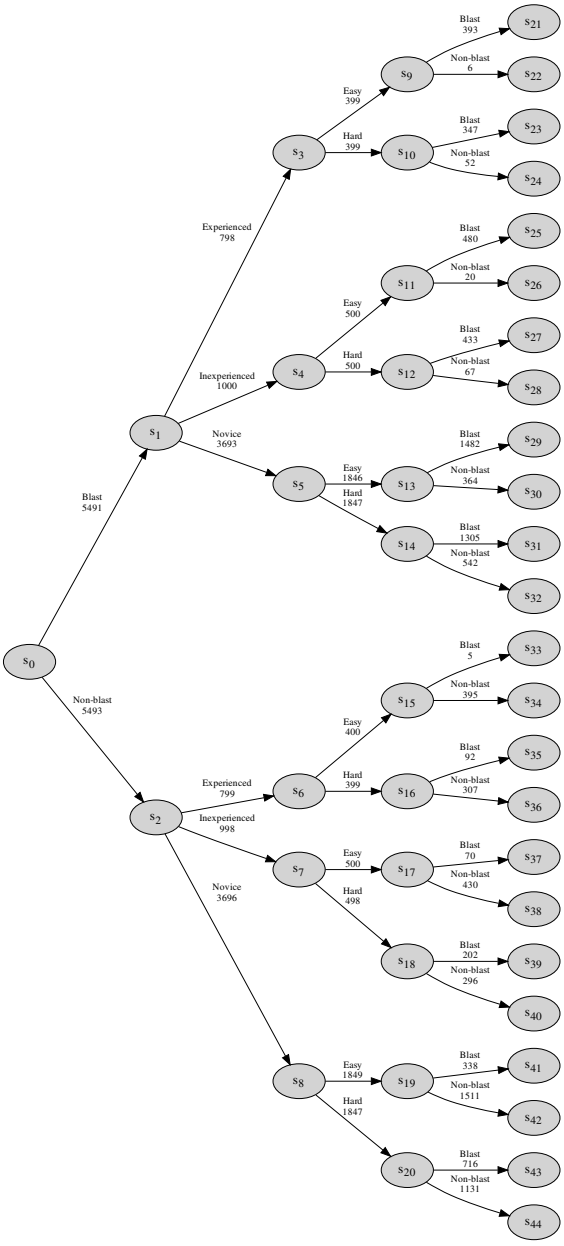


Figure 3.8: Event tree of the medical decision-making example using `cegpy`

Any paths which logically belong in the event tree description of the process but are absent from the dataset due to sampling limitations can be manually added by using the *sampling_zero_paths* argument when initialising the EventTree object. Additionally, not all missing values in the dataset are structurally missing; to differentiate between the two, we can assign different labels to the structural and

sampling missing values.

The Staged Tree

To create a staged tree from our dataset, we can initialise a StagedTree object with the dataset as input, without needing to first initialise an EventTree object. Before creating the staged tree, it is important to identify the stages in the event tree, which can be accomplished by running the AHC algorithm within the StagedTree object. By default, the hyperstage will contain any vertices that have the same number of outgoing edges and have the same edge labels. Also by default, the prior is set through alpha uniformly distributed phantom samples, starting at the root. Either the prior itself or the alpha parameter can be specified, with the default value of alpha being the maximum number of categories that a variable has [Neapolitan, 2003]. The priors and posteriors are saved as fractions to maintain accuracy during the iterative calculations.

```
from cegpy import StagedTree
st = StagedTree(df)
print('default hyperstage:',st._create_default_hyperstage())
print('default alpha:',st._calculate_default_alpha())
print('default prior:',st._create_default_prior(st._calculate_default_alpha()))
```

```
default hyperstage: [['s0', 's9', 's10', 's11', 's12', 's13', 's14', 's15', 's16',
's17', 's18', 's19', 's20'], ['s1', 's2'], ['s3', 's4', 's5', 's6', 's7', 's8']]
default alpha: 3
default prior: [[Fraction(3, 2), Fraction(3, 2)], [Fraction(1, 2), Fraction(1, 2),
Fraction(1, 2)], [Fraction(1, 2), Fraction(1, 2), Fraction(1, 2)], [Fraction(1, 4),
Fraction(1, 4)], [Fraction(1, 4), Fraction(1, 4)], [Fraction(1, 4), Fraction(1, 4)],
[Fraction(1, 4), Fraction(1, 4)], [Fraction(1, 4), Fraction(1, 4)], [Fraction(1, 4),
Fraction(1, 4)], [Fraction(1, 8), Fraction(1, 8)], [Fraction(1, 8), Fraction(1, 8)],
[Fraction(1, 8), Fraction(1, 8)], [Fraction(1, 8), Fraction(1, 8)], [Fraction(1, 8),
Fraction(1, 8)], [Fraction(1, 8), Fraction(1, 8)], [Fraction(1, 8), Fraction(1, 8)],
[Fraction(1, 8), Fraction(1, 8)], [Fraction(1, 8), Fraction(1, 8)], [Fraction(1, 8),
Fraction(1, 8)], [Fraction(1, 8), Fraction(1, 8)], [Fraction(1, 8), Fraction(1, 8)]]
```

Using the code below, we can run the AHC algorithm with the default settings described above and generate the corresponding staged tree figure, as illustrated in Figure 3.9.

```
st.calculate_AHC_transitions()
st.create_figure()
```

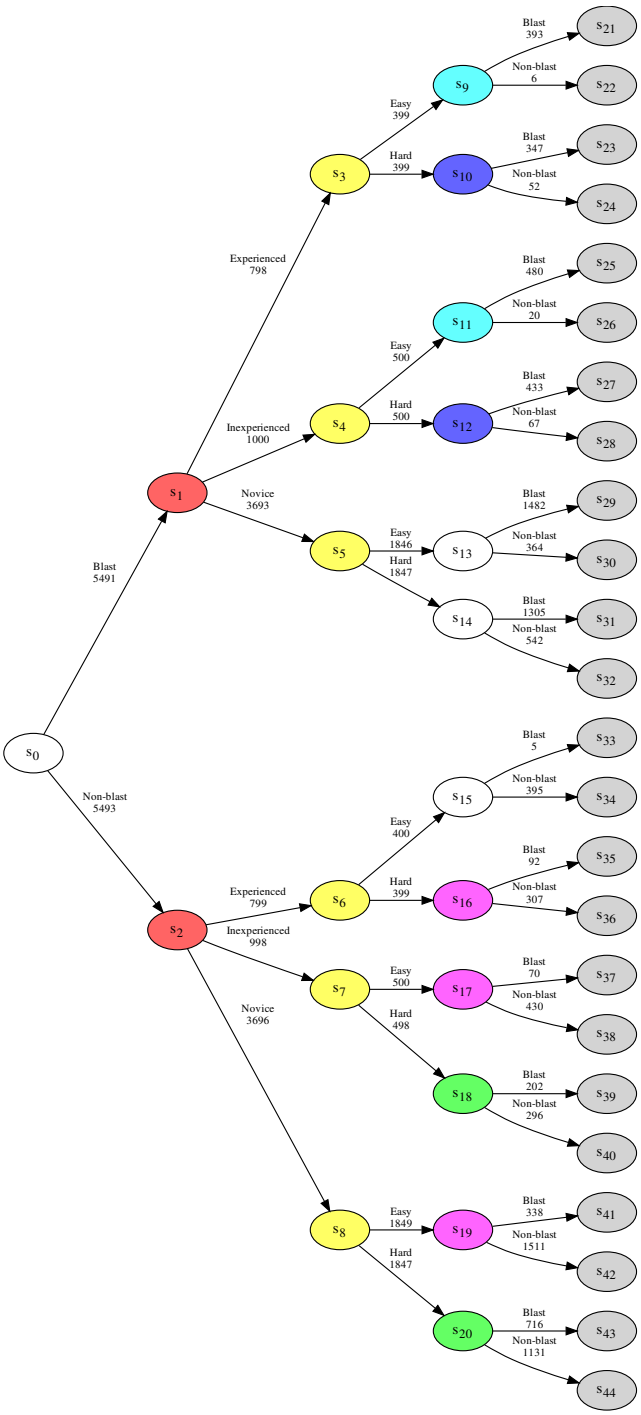


Figure 3.9: Staged tree of the medical decision making example.

The CEG

After identifying the stages by running the AHC algorithm on the StagedTree object, we can create a ChainEventGraph object that takes the StagedTree object as input. By using the StagedTree object, the ChainEventGraph object can generate the CEG figure using the code below, which is illustrated in Figure 3.10.

```
from cegpy import ChainEventGraph
ceg = ChainEventGraph(st)
ceg.create_figure()
```

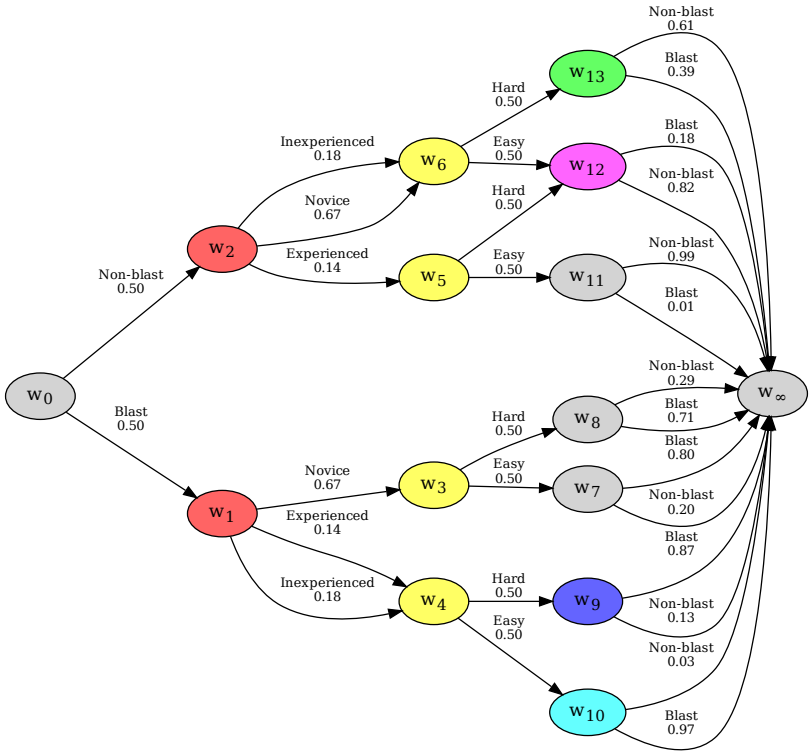


Figure 3.10: CEG of the medical decision making example.

3.6.3 Summary

The Python package cegpy is a useful tool, designed for modelling with staged trees and CEGs, which includes Bayesian model selection and probability propagation capabilities. It is the first CEG and staged tree package in Python and the first package to model non-stratified CEGs, providing additional capabilities for applied statisticians.

Further development is ongoing to enable users to directly specify an event tree, staged tree or CEG structure – with colouring and possibly, with parameters – in the `cegpy` package. Of course, learning algorithms cannot be used due to the absence of data but it would be beneficial for visualisation and evidence propagation. We are currently looking into adding this functionality by directly importing graphs specified using the DOT language used by `GraphViz`.

Currently, the AHC algorithm is the only supported Bayesian model selection algorithm, but methodological developments from the base package have led to work in Strong and Smith [2022b] and Strong and Smith [2022a]; details of these developments can be seen in Chapters 4 and 5.

Chapter 4

Bayesian model averaging of Chain Event Graphs

Chapter 3 detailed Bayesian CEG model selection based on a Dirichlet distribution over a fixed tree; such methods were implemented in Section 3.6. These approaches, however, have been focused on finding a single CEG, in a Bayesian context the MAP CEG. This ignores any uncertainty about the independence statements learned in the model selection and can therefore lead to overconfident and, sometimes, spurious inferences. In this chapter, we outline the use of Bayesian model averaging (BMA) techniques to incorporate model uncertainty.

We begin this chapter by further motivating the need for handling uncertainty around the independence statements learned through model selection in Section 4.1. Then, in Section 4.2, we introduce the concept of BMA. In Section 4.2.1, we discuss how Occam's window is used as part of this methodology to support reducing the number of models we consider and why this is important. In Section 4.3, we show how BMA can be applied to CEGs. In Section 4.4.1, we propose the use of a sampling algorithm to address these issues and define one potential algorithm, wr-HAC. In Section 4.5, we demonstrate the benefits of BMA through a worked example. The material in this chapter is based on Strong and Smith [2022b].

4.1 Introduction

To convey the issues with only using a single CEG, we will begin with an illustrative example.

Example 32 (Motivating Example) *Suppose we have a sequence of two events (A and B) with two outcomes, as shown in Figure 4.1. Observations are given by*

the edge counts. Assuming the hyperstage $\{\{s_0\},\{s_1,s_2\}\}$ (see Section 3.4.1), the model selection in this instance is between two CEGs: when Event B is independent of Event A (known herein as the "independence model"), and when it is not (the "saturated model"). These scenarios are shown in Figures 4.2 and 4.3 respectively. Suppose we set a prior as recommended by Neapolitan [2003], detailed in Section 3.3.1. When performing model selection, we use the BDeu score (see Section 3.3.1). The BDeu scores will be evaluated and the model with the staging that has the highest BDeu score will be chosen as the MAP model.

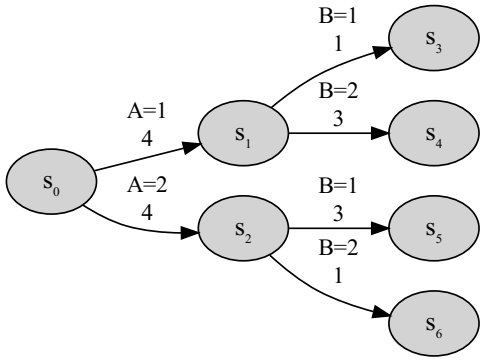


Figure 4.1: Event tree showing a sequence of two events– A and B– with two outcomes. The counts on the edges show the data.

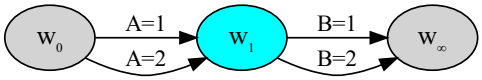


Figure 4.2: The 'independence' model: CEG where Event B is independent of Event A.

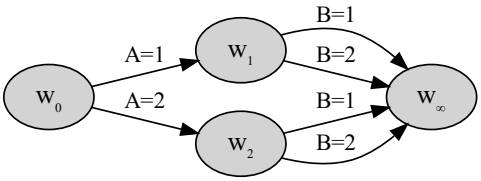


Figure 4.3: The 'saturated' model: CEG where Event B is not independent of Event A.

The log BDeu score for the independence model is -9.88 and for the saturated model is -9.43. In this instance, the 'saturated' model is the MAP model (Figure 4.3).

However, the difference between the model scores is quite small: a BF of 1.58. Table 3.1 from Kass and Raftery [1995] provides an interpretation of this BF of 'Not worth more than a bare mention'. This shows that, despite the 'saturated' CEG best representing the data, there is not much difference in the strength of evidence between the two models. Therefore, by using this single model, the uncertainty of the strength of the staging is being lost. Using the MAP model as a focus of inferences is also

not robust as small variations in the observed data can lead to a different model.

This example may seem contrived due to the small numbers observed on the florets. Whilst that is true, it is highly feasible that a tree like that existing in Figure 4.1 could exist as a subtree of a much larger tree for which a CEG is learnt. In a larger tree, there will also be a larger set of possible models, obfuscating the best model for representing the process.

In summary, using a single selected CEG, such as the MAP model, often provides helpful insights into the underlying data-generating process when that candidate has a high posterior probability. However, when this is not the case, focusing only on this model will lead to overconfident and sometimes spurious inferences. This occurs when there are many high-scoring models with non-negligible probabilities, a phenomenon that is common if the size of the model space is much larger than the number of data points. This is a typical scenario in all but the simplest of settings [Tian et al., 2010], including for CEGs. More recently, non-Bayesian approaches to learning the staging structure based on clustering have been used [Carli et al., 2020]. However, these methods solely focus on obtaining a single CEG that maximises some score, so are susceptible to the same problems as those outlined above.

These realisations led us to conclude that in many circumstances it is not ideal to select a single model and focus all inferences around this. Instead, we should search for an *optimal set* of the highest-scoring models. Suppose we believe that a data-generating process can be described by models in a particular chosen space. From a Bayesian perspective, the principled way of performing predictive inference is to simply use the posterior model probabilities to inform the optimal set.

4.2 Bayesian Model Averaging

In this section, we introduce BMA, based on Madigan and Raftery [1994] and Hinne et al. [2020], as a method to address the issues raised in Section 4.1. BMA involves using a collection of models to describe the data-generating process instead of a single model. This is a naturally Bayesian way to approach the problem of model selection by, instead of using the best model for any analysis, considering all possible models weighted by their plausibility. This allows for uncertainty about the parameters and the model space.

Because of this, BMA removes overconfidence, with a single model only taken in the limit. The output of BMA is more robust than that of a single model, with a small change in the observed data leading to a similar distribution of models instead

of a potentially entirely different model. Importantly, in the instance where there is a single model that is a much better representation of the process than any of the other models, the BMA will be similar to that of a single model.

More formally, when the focus of our inference is on Φ (the conditional transition probabilities), given data $\boldsymbol{y}$, models $M_k \in \mathcal{S}$, $k \in \{1, \ldots, N\}$ and model space $\mathcal{S}$, the BMA is given by:

$$p(\Phi|\boldsymbol{y}) = \sum_{k=1}^{N} p(\Phi|\boldsymbol{y}, M_k)p(M_k|\boldsymbol{y}). \tag{4.1}$$

This shows that the prediction is a weighted average of each of the K competing models. Here, $p(\Phi|\boldsymbol{y}, M_k)$ is the posterior probability for model M_k and $p(M_k|\boldsymbol{y})$ is the posterior probability of model M_k given by,

$$p(M_k|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|M_k)p(M_k)}{\sum_{i=1}^{N} p(\boldsymbol{y}|M_i)p(M_i)}. \tag{4.2}$$

We can represent the posterior odds of two models as follows

$$\frac{p(M_k|\boldsymbol{y})}{p(M_l|\boldsymbol{y})} = \frac{p(\boldsymbol{y}|M_k)}{p(\boldsymbol{y}|M_l)} \times \frac{p(M_k)}{p(M_l)}. \tag{4.3}$$

Using the BF, $BF(M_k, M_l) = \frac{p(\boldsymbol{y}|M_k)}{p(\boldsymbol{y}|M_l)}$, we can represent Equation (4.2) as follows:

$$p(M_k|\boldsymbol{y}) = \frac{BF(M_k, M_1)p(M_k)}{\sum_{i=1}^{N} BF(M_i, M_1)p(M_i)} \tag{4.4}$$

where $BF(M_k, M_1)$ is the BF comparing model k to any other model denoted here as 1. If a uniform prior is set, $p(M_i) = p(M_j)$ for all $i, j \in \{1, \ldots, N\}$, then

$$p(M_k|y) = \frac{BF(M_k, M_1)}{\sum_{i=1}^{N} BF(M_i, M_1)}. \tag{4.5}$$

Therefore, we can calculate the weights from the BFs of each model compared to a single model.

Example 33 (Motivating example continued) *We return to the motivating example in Section 4.1, but instead of performing model selection to choose a single model, we perform BMA. Denote M_1 as the 'saturated' model (Figure 4.3) and M_2 as the 'independence' model (Figure 4.2). Assuming a uniform prior over the set of models, we can calculate each model's posterior probability using Equation (4.5), with the following result:*

$$p(M_1|y) = 0.61, \quad p(M_2|y) = 0.39. \tag{4.6}$$

4.2.1 Occam's Window

One of the drawbacks of BMA is that performing model averaging over all models is often an intractable problem due to the large number of potential models [Madigan and Raftery, 1994]. Whilst it is important that a large number of possible models are considered to account for different data-generating processes, when performing model averaging, it is necessary for the average to be over a smaller set of models. Occam's window helps us to keep in the frame only the most supported models, discarding those which have been discredited. This is based on two steps: first, we discard any of the models that are at least β times less likely than the best performing model[1]. We denote this set of models $\mathcal{S}'$ and is given by:

$$\mathcal{S}' = \left\{ M_k : M_k \in \mathcal{S} \wedge \frac{\max_{M_l \in \mathcal{S}} p(M_l|y)}{p(M_k|y)} < \beta \right\}. \tag{4.7}$$

Secondly, we perform a step based around Occam's razor in which, given multiple models with the same amount of evidence, the simplest is preferred. This means we discard any models for which there exists a nested, more likely model. In Equation (4.8), we denote $M_l \subset M_k$ if M_l is a nested model of M_K.

$$\mathcal{R} = \left\{ M_k : \exists M_l \in \mathcal{S}', M_l \subset M_k \wedge \frac{p(M_k|y)}{p(M_l|y)} < 1 \right\} \tag{4.8}$$

Here, $\mathcal{R}$ is the set of models that should be removed in the Occam's razor step.

$$\hat{\mathcal{S}} = \mathcal{S}' \backslash \mathcal{R} \tag{4.9}$$

This leaves a set of models, $\hat{\mathcal{S}}$, which we will refer to as the *well-performing* models.

When dealing with interpretable models, Occam's window is not just an approximation. It also effectively enables us to focus on good explanatory models and discard the rest. This is vital when there might be many poorly fitting models in the space *a priori*; although none of these explains the process well, the residual probability on these remains large after sampling, which can blur the posterior image until we are able to gather enormous amounts of data. This will be the case when, for example, we do not have the time or expertise to forensically set priors on models

[1] A standard choice of β is 20 [Fragoso et al., 2018] aligning with the popular 0.05 cutoff for p-values.

that *a priori* should be assigned a small probability.

An alternative to Occam's window is to use k-best models and perform model averaging over these [Tian et al., 2010]. However, in this work we decided to use Occam's window where the number of models averaged over is an indicator of the uncertainty over the choice of model.

4.3 Model Averaging for CEGs

BMA provides an excellent method in which to measure the robustness of explanations with respect to model uncertainty, especially when each scored model has an associated explanation to accompany it, as is the case for CEGs. When explanations are shared across many high-scoring models, inferences are more secure than when such explanations only apply to a single, highest-scoring model. In this section, we will detail how BMA can be used for CEGs.

4.3.1 Nested Models

Of course, except in small model spaces, it is often impossible to average over all possible stagings of CEGs. As the number of CEGs grows super-exponentially with the number of events [Silander and Leong, 2013], it is important that we reduce the number of models in the BMA to simplify model interpretation. We will do this using Occam's window, as described in Section 4.2.1. A key part of Occam's window is its Occam's razor step, in which complex models for which there are nested simpler models that have better marginal likelihoods are removed.

Here, we define a CEG as *nested* in another CEG when it can be obtained by combining its stages, as detailed in Section 3.4.4. More precisely, the stages of the contained CEG are obtained as unions of the stages of the containing CEG. This means all CEGs are nested in the CEG with all situations in different stages and the CEG with all situations– in the same hyperstage– in the same stage is nested in all other CEGs. This nesting creates a partial order over the class of CEGs.

This means that, when considering a simple tree like the one shown in Figure 4.1, if the model with the higher marginal likelihood was the simpler model with the two situations in the same stage (Figure 4.2), that would be the only model in the Bayesian model average after applying Occam's window.

4.3.2 Unique Representation

In BMA, when setting a uniform prior over a set of models, it is important that no models are represented twice. For BNs, a uniform prior is often set over the Markov equivalence class [Madigan et al., 1996]. While the operations that define the possible ways to transverse the equivalence class of CEGs is known [Görgen and Smith, 2018], the cardinality of each equivalence class remains an open research problem. Therefore, in this chapter, we focus on the situation in which the tree has already been directly elicited. This has the advantage that there only exists a single model in each equivalence class.

4.3.3 Independent Hyperset Staging

The hyperstage determines the allowed stagings in a CEG, as detailed in Section 3.4.1, and is an important part of model selection. As shown in Section 3.3, the marginal likelihood of a CEG is a product of the marginal likelihood of its stages. Therefore, when the hyperstage is a partition of the set of situations, the marginal likelihood of stages within each hyperset can be examined independently. Therefore, the issue of determining the weights for each model can be reformulated as finding the different weights for each hyperset staging, with the model weight given by the product of the different staging weights.

This can be shown by rewriting Equation (3.4), by grouping the stagings by their hyperset as follows:

$$p(\boldsymbol{y}|\mathcal{C}) = \prod_{H_i \in \boldsymbol{H}} \left[\prod_{j \in H_i} \left[\frac{\Gamma(\overline{\boldsymbol{\alpha}}_j)}{\Gamma(\overline{\boldsymbol{\alpha}}^*_j)} \prod_{k=1}^{k_j} \frac{\Gamma(\alpha^*_{jk})}{\Gamma(\alpha_{jk})} \right] \right]. \tag{4.10}$$

This rewriting is done by grouping the stages in the marginal likelihood calculation, by what hyperset, H_i, they are in.

Example 34 (Hyperset Setting) *Suppose we have 3 binary events as given in Figure 4.4 with the hyperstage $\{\{s_0\}, \{s_1, s_2\}, \{s_3, s_4\}\}$ shown by the coloured nodes. There are one, two and two different ways of staging each hyperset respectively. Therefore, there are four different ways the tree can be staged. As shown in Equation (4.10), the marginal likelihood of each of these models can also be given by the product of the marginal likelihood of each of the hypersets.*

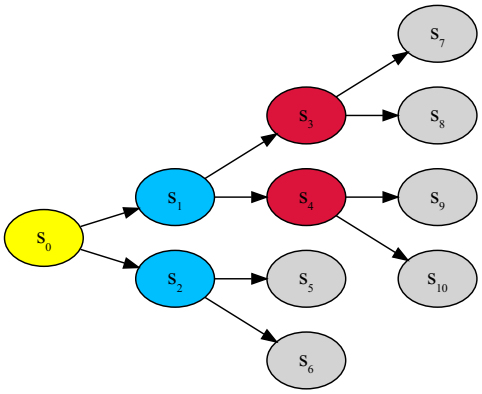


Figure 4.4: Tree showing 3 binary events with colours representing the hyperstage structure and the leaves in grey.

This means different potential stagings of a hyperset can be studied independently and any differences in that hyperstage can be seen in isolation. Staging each hyperstage independently is consistent with the Occam's razor step of removing more complicated models which have a worse marginal likelihood. Suppose you have two potential stagings of a hyperset: one with a larger marginal likelihood which corresponds to a simpler model than the other. If we consider any two models which have the same stagings except for this hyperset, the simpler hyperset with the larger marginal likelihood will always have a higher posterior probability than the alternative staging; this means we would remove the alternative model from BMA as it would be more complex with a worse marginal likelihood. Therefore, to perform BMA, we only need to consider the staging of each hyperset.

4.3.4 Measure of Separation

It is critical to note that, as CEGs are an interpretable class of models, using the hyperstage means that we only consider certain partitions of the situations into stages. As each partition corresponds to asserting probability distributions on corresponding florets, CEGs with 'close' partitions will have similar interpretations.

In order to quantify the closeness between two stagings of a hyperset, we define a measure of separation based on the partition of sets within our hyperstage. For model k, we define its staging, $S_{i,k}$, of hyperset $H_i \in \boldsymbol{H}$. We define $S_{i,1} \preceq S_{i,2}$ if all stagings in $S_{i,1}$ are subsets of stages in $S_{i,2}$ with $S_{i,1}$ defined as a *refinement* of $S_{i,2}$ and $S_{i,2}$ a *coarsening* of $S_{i,1}$. Relating this to the nested terminology in Section 4.3.1, a more refined staging is nested in a coarser staging.

Using this relation, we further define the coarsest intersection, $S_{i,1} \wedge S_{i,2}$, to be the coarsest staging of H_i, such that all stages in $S_{i,1} \wedge S_{i,2}$ are contained

in one staging in $S_{i,1}$ and one staging in $S_{i,2}$. For example, if $S_{i,1}$ is a coarsening of $S_{i,2}$, then $S_{i,1} \wedge S_{i,2} = S_{i,1}$. Also, if for example we had the hyperstage $H_i = \{s_1, s_2, s_3\}$ and stagings $S_{i,1} = \{\{s_1, s_2\}, \{s_3\}\}$ and $S_{i,2} = \{\{s_1\}, \{s_2, s_3\}\}$, then $S_{i,1} \wedge S_{i,2} = \{\{s_1\}, \{s_2\}, \{s_3\}\}$. The stages that exist in the coarsest intersection gives the situations that are in the same stage in both of the models, providing a greatest lower bound of two stagings.

Similarly, the most refined union between two stagings, $S_{i,1} \vee S_{i,2}$, can be defined to be the most refined staging of H_i, such that all stages in $S_{i,1}$ and $S_{i,2}$ are subsets of stages in $S_{i,1} \wedge S_{i,2}$. The stages that exist in the most refined union gives the situations that are in different stages in both of the models, providing a lowest upper bound of two stagings. The coarsest intersection and the most refined union can be used to determine which inferences are most secure.

There are various ways of defining a separation measure on a staging of a hyperset: we define the following separation measure between two partitions $S_{i,1}$ and $S_{i,2}$ by

$$\begin{aligned} d(S_{i,1}, S_{i,2}) =& [\#(S_{i,1} \wedge S_{i,2}) - \#(S_{i,1})] \\ &+ [\#(S_{i,1} \wedge S_{i,2}) - \#(S_{i,2})] \\ =& 2 \times \#(S_{i,1} \wedge S_{i,2}) - \#(S_{i,1}) - \#(S_{i,2}). \end{aligned} \tag{4.11}$$

Here $\#(S_{i,k})$ denotes the number of stages in $S_{i,k}$. This measure of separation is based on the topology induced by AHC, where stagings are traversed by merging stagings together, as discussed in Section 3.4.4. This separation measure is the sum of the mergings of stages which have to take place from $S_{i,1} \wedge S_{i,2}$ to reach each of the stagings $S_{i,1}$ and $S_{i,2}$. Note that

$$d(S_{i,1}, S_{i,2}) = 0 \iff S_{i,1} = S_{i,2}. \tag{4.12}$$

If $d(S_{i,1}, S_{i,2}) = 1$, then the coarser of $S_{i,1}$ and $S_{i,2}$ can reach the other by merging together two of its stages. It also follows that if we define $S = S_{i,1} \wedge S_{i,2}$, then $d(S_{i,1}, S_{i,2}) = d(S_{i,1}, S) + d(S_{i,2}, S)$.

Example 35 (Hyperset Setting Continued) *Continuing the example, we have stagings* $S_{i,1} = \{\{s_1, s_2\}, \{s_3\}\}$ *and* $S_{i,2} = \{\{s_1\}, \{s_2, s_3\}\}$. *Here,* $d(S_{i,1}, S_{i,2}) = 2$ *as single merging is needed to reach each model's staging from* $S_{i,1} \wedge S_{i,2}$.

The partial ordering here introduces an ordering on the stagings, creating a lattice which can be visualised using a Hesse diagram. In this lattice, the different

stagings represent nodes with a partial order between two stagings if one is a refinement of the other[2]. A Hesse diagram showing the different possible stagings of a hyperset with 4 situations in it is shown in Figure 4.5.

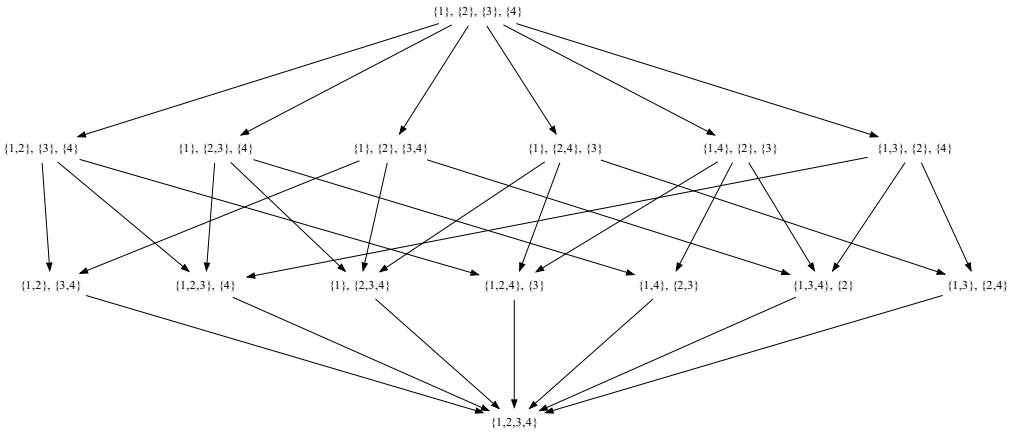


Figure 4.5: Hesse diagram of the possible stagings of a hyperset with 4 situations, denoted $(1, 2, 3, 4)$ instead of (s_1, s_2, s_3, s_4) for readability, with arrows showing the partial order created by merging situations.

We can use the Hesse diagram to visualise the uncertainty in the BMA using the greatest lower bound and the smallest upper bound which are represented by the closest common ancestor and descendent respectively. To illustrate this, see the following example.

Example 36 (Hesse diagrams for bounds) *Suppose we are interested in staging a hyperset with 4 situations. For ease of visualisation, these situations are denoted* $(1, 2, 3, 4)$ *instead of* (s_1, s_2, s_3, s_4)*. We will consider a few cases:*

[2]As AHC starts at the coarsest model then considers refinements to increases its BF, the lattice defined by the partial order of refinements gives the possible next-step mergings which can be visualised in a Hesse diagram.

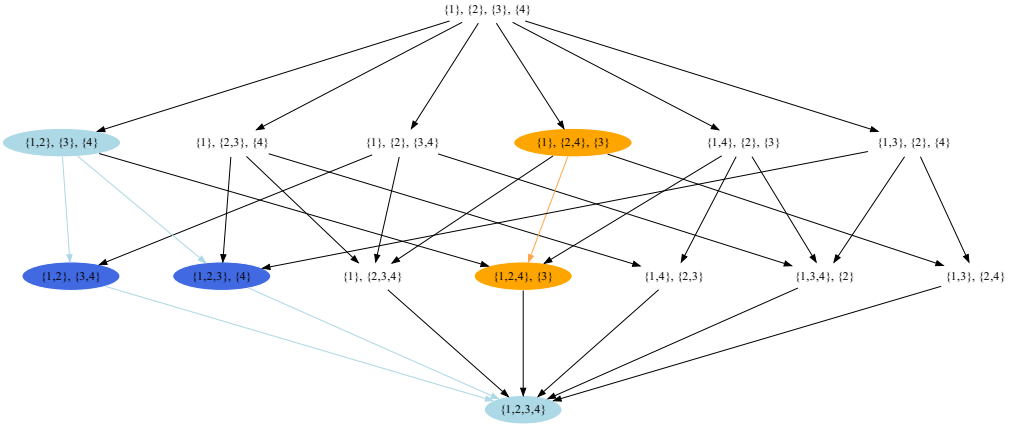


Figure 4.6: Hesse diagram of the possible stagings of a hyperset with 4 situations. This diagram is coloured blue/orange and light blue/orange to show the stagings and their bounds respectively for each example. Edges coloured light blue denote the paths to the situations' nearest common ancestors and descendants.

First, assume that we have the stagings $\{1,2\},\{3,4\}$ *and* $\{1,2,3\},\{4\}$ *shown in dark blue in Figure 4.6; their coarsest intersection and most refined union are given in light blue.*

Next, assume that we have the stagings $\{1,2,4\},\{3\}$ *and* $\{1\},\{2,4\},\{3\}$ *shown in orange in Figure 4.6. Note here that, as one is a refinement of the other, their coarsest intersection and most refined union are the coarser and more refined of the two respectively.*

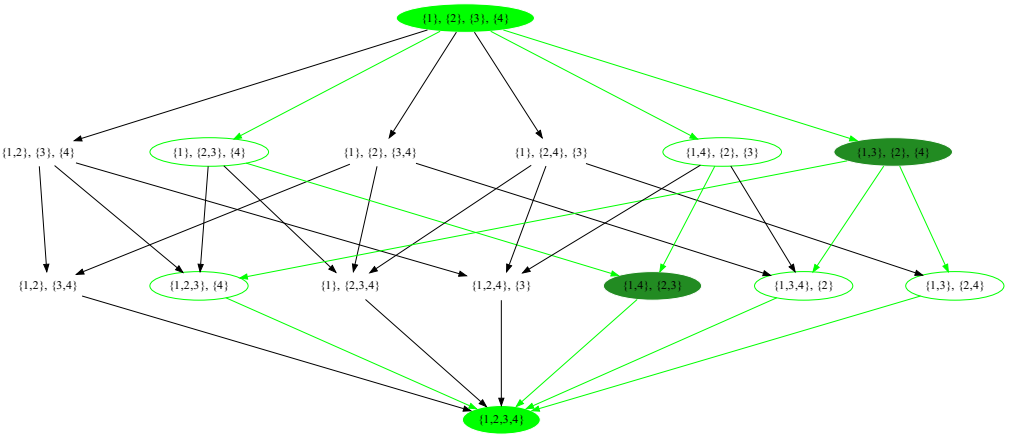


Figure 4.7: Hesse diagram of the possible stagings of a hyperset with 4 situations. This diagram is coloured green and light green to show the stagings and their bounds respectively. Edges and vertex borders coloured light green denote the paths to the situations' nearest common ancestors and descendants.

Finally, assume that we have the stagings $\{1,4\},\{2,3\}$ *and* $\{1,3\},\{2\},\{4\}$ *shown in dark green in Figure 4.7. Their coarsest intersection and most refined union are given in light green. In this example, there are no situations that are always together or apart so the bounds on the staging includes the whole class of models.*

4.4 Sampling the Model Space

As the number of possible staged trees grows super-exponentially, even *calculating* all the posterior probabilities for each model in the space of CEGs becomes an intractable problem. We propose the following approach to address this issue. First, take a sample of size n from the model space, while aiming to include the models with the highest posterior probability in the sample. Then, apply Occam's window to obtain an approximation of a set of well-performing models. The aim of our approach is to obtain a set of models that are a good approximation of the set of models that would lie within Occam's window if all the posterior probabilities could be calculated. This set of models provides a good narrative, with a few alternatives, of the process being modelled. Most importantly, this approximation will allow for quantification of model uncertainty, and therefore certainty in the model's independence statements.

More formally, to approximate a BMA, $\boldsymbol{\pi}$, of a set of models, $\mathcal{S}$, we take independent samples from the model space $\hat{\boldsymbol{S}}_n = \{S_1, \ldots, S_n\}$ which are used to calculate an estimator of the BMA, $\hat{\boldsymbol{\pi}}_n$, by applying Equation (4.5) to the unique set of models sampled, denoted $U(\hat{\boldsymbol{S}}_n)$. Here, $\hat{\boldsymbol{\pi}}_n \in [0,1]^M$, where M is the number of models in $\mathcal{S}$. Note that almost all of these dimensions will be zero as they will not be in the sample $\hat{\boldsymbol{S}}_n$.

We would like our sampler to be a consistent estimator.

Definition 37 *Consistency: An estimator* $\hat{\boldsymbol{\pi}}_n$ *of parameter* $\boldsymbol{\pi}$ *is consistent, if it converges in probability to the true value of the parameter i.e. for all* $\epsilon > 0$*:*

$$\lim_{n\to\infty} (P(\|\hat{\boldsymbol{\pi}}_n - \boldsymbol{\pi}\| > \epsilon)) = 0. \tag{4.13}$$

Theorem 38 *An estimator* $\hat{\boldsymbol{\pi}}_n$ *of* $\boldsymbol{\pi}$*, a BMA over a set of models* $\mathcal{S}$*, obtained by calculating the posterior probabilities of models reached by an independent sampler,* $\hat{\boldsymbol{S}}_n$*, is consistent if for all* $M_i \in \mathcal{S}$*,* $P(S_j = M_i) = p_i > 0$*.*

Proof. The vector $\boldsymbol{\pi}$ has components given by the posterior probabilities of each of the models in $\mathcal{S}$, calculable through a ratio of BFs. Estimator $\hat{\boldsymbol{\pi}}_n$ is obtained by

calculating the ratio of BFs of the models sampled $\hat{\boldsymbol{S}}_n = \{S_1, \ldots, S_n\}$. Therefore, if $\hat{\boldsymbol{S}}_n = \mathcal{S}$ then $\hat{\boldsymbol{\pi}}_n = \boldsymbol{\pi}$,

$$P(M_i \in \hat{\boldsymbol{S}}_n) = 1 - (1 - p_i)^n.$$

Therefore, as for all $M_i \in \mathcal{S}$, $p_i = P(S_n = m_i) > 0$,

$$\lim_{n\to\infty} P(M_i \in \hat{\boldsymbol{S}}_n) = 1.$$

Therefore $\lim_{n\to\infty} \hat{\boldsymbol{S}}_n = \mathcal{S}$ and

$$\lim_{n\to\infty} (P\|\hat{\boldsymbol{\pi}}_n - \boldsymbol{\pi}\| > \epsilon) = 0.$$

■

Corollary 39 *An estimator $\hat{\boldsymbol{\pi}}_n^*$ of an Occam's window BMA, $\boldsymbol{\pi}^*$, over a set of models $\mathcal{S}$ obtained by applying Occam's window to an independent sampler, $\hat{\boldsymbol{S}}_n$ of potential models, is consistent if for all $M_i \in \mathcal{S}^*$, where $\mathcal{S}^*$ is the set of models in the Occam's window BMA set, $p(S_j = M_i) > 0$.*

Proof. We proceed in exactly the same way as in Theorem 38. The components of $\boldsymbol{\pi}^*$ are the posterior probability of each of the models in $\mathcal{S}^*$, calculable through a ratio of BFs with Occam's window applied. Here, $\boldsymbol{\pi}^*$ is 0 in each dimension, representing a model removed by the Occam's window. Estimator $\hat{\boldsymbol{\pi}}_n$ is obtained by calculating the ratio of BFs of the sampled models $\hat{\boldsymbol{S}}_n = \{S_1, \ldots, S_n\}$, then applying Occam's window. Therefore, if $\mathcal{S}^* \subseteq \hat{\boldsymbol{S}}_n$, then $\hat{\boldsymbol{\pi}}_n^* = \boldsymbol{\pi}^*$,

$$P(M_i \in \hat{\boldsymbol{S}}_n) = 1 - (1 - p_i)^n.$$

Therefore, as for all $M_i \in \mathcal{S}^*$, $p_i = P(S_n = M_i) > 0$

$$\lim_{n\to\infty} P(M_i \in \hat{\boldsymbol{S}}_n) = 1.$$

Therefore, $\mathcal{S}^* \subseteq \lim_{n\to\infty} \hat{\boldsymbol{S}}_n$ and

$$\lim_{n\to\infty} (P\|\hat{\boldsymbol{\pi}}_n - \boldsymbol{\pi}\| > \epsilon) = 0.$$

■

This means that our sampler only needs to sample models that will remain in the Occam's window model average. Although these are of course not known *a priori*, features in the marginal likelihood can be used to try to centre the activities

of any sampler so that these do not sample heavily in likely irrelevant regions of the parameter space.

The partitioning of the model search into each hyperset, as mentioned in Section 4.3.3, enables us to prioritise computational resources. This means that for hypersets with many more elements and therefore many more possible stagings, more computational resources can be used to sample this space. The set of CEGs that we are model averaging over can then be obtained by taking all possible combinations of well-performing stagings of each hyperset.

4.4.1 wr-HAC Algorithm

In this chapter, for simplicity, we content ourselves with comparing the performance of model averaging against MAP estimation for searching across explanations using a very simple – although, as far as we know, novel – search algorithm. This is because, as long as the search algorithm is able to satisfy the properties to make it a consistent estimator, the choice of search algorithm is only important for matters of computational efficiency. We show that, even with this naive method, our search performs surprisingly well for the purposes of BMA.

Strong and Smith [2022b] proposed a weighted version of the AHC (also known as HAC) algorithm, weighted hierarchical agglomerative clustering (w-HAC). Instead of being a greedy search algorithm, as in AHC, w-HAC is a randomised algorithm, where the probability of merging is weighted by the relative BFs of the potential mergings. Under w-HAC, the probability of two stages, u_j and u_k, merging, in hyperset H_i, is given in Equation (4.14):

$$p(u_j, u_k) = \frac{BF(S_{i,j\oplus k}, S_{i,1})}{\sum_{k,l\in S_{i,1}} BF(S_{i,k\oplus l}, S_{i,1}))}. \tag{4.14}$$

Here, $S_{i,1}$ is the staging from which the merges are being considered from and $BF(S_{i,j\oplus k}, S_{i,1})$ is the BF comparing when stages j and k are merged to when they are not.

However, w-HAC does not necessarily provide a consistant estimator of the BMA, due to it stopping when there is no merging that will increase the BF. This means that using w-HAC to perform BMA could miss nested models which have lower marginal likelihood but which would remain in the Occam's window.

Here, we extend w-HAC into the weighted random hierarchical agglomerative clustering algorithm (wr-HAC). This introduces a probability γ that, given there is no combination of stages that would lead to an increase in the BF, a merging still takes place based on the weighting in Equation (4.14). This means that the

algorithm samples models which are worse-performing and simpler; this satisfies the conditions of Corollary 4.4 and makes it a consistent estimator.

Due to the stopping criteria for wr-HAC – only possibly stopping if there are no possible situations left to be merged or if none of the potential mergers would increase the BF – we know that our set of solutions will satisfy a weaker version of Occam's razor: it will not include a model that is less probable than a one-nested simpler model. This is because, due to the latter condition, if a simpler nested model existed, there would be a merged model with a positive log(BF). Therefore, a merging would occur.

For explanatory modelling, it is important to obtain the most likely model. For CEGs, a method, other than exhaustive search, for finding the MAP model is still an open research question, with AHC currently providing the best estimates. As AHC is simply a greedy version of wr-HAC, we hypothesise that the set of models outputted by iterations of wr-HAC will contain AHC's estimation of the MAP model alongside other high-scoring models, as well as potentially avoiding local maxima. We note that, while K-means for different values of K, could be used to give a set of models to perform model averaging on, this would be a less promising choice since its output has been shown to perform worse than AHC as an estimation of the MAP model [Silander and Leong, 2013]. Therefore, the BMA is less likely to include the model with the highest posterior probability in its BMA making the model uncertainty misspecified.

Input : Event tree $\mathcal{T}$ its associated hyperstage $\boldsymbol{H}$, data $\mathbf{y}$ and root equivalent sample size $\bar{\boldsymbol{\alpha}}_0$.

Output: A CEG and its associated staging and log marginal likelihood score.

Initialise *data*, y_i for each situation s_i in $\mathcal{T}$ from $\mathbf{y}$.

Initialise *priors*, $\boldsymbol{\alpha}_i$ for each situation s_i in $\mathcal{T}$ from $\bar{\boldsymbol{\alpha}}_0$ through mass conservation.

Initialise a *stage* for each situation s_i in $\mathcal{T}$

Set *score* as the log marginal likelihood score given in Equation (3.7).

Set *indicator* $\leftarrow 1$.

while *indicator* $\neq 0$ **do**

  if *There is a single stage in every hyperstage* **then**

    indicator $\leftarrow 0$

  end

  for *every pair of stages in stages in the same hyperstage* **do**

    Calculate the $\log BF$ as given in Equation (3.10) comparing the structures of merging the pair to keeping them apart, all other stages being equal.

  end

  if *There exists a calculated logBF* ≥ 0 *or with probability* γ **then**

    choose a pair u_i and u_j weighted by their BF as in Equation (4.14)

    for *pair* u_i *and* u_j **do**

      score $\leftarrow$ *score* $+ \log BF(u_i, u_j)$

      Update *stages* to add stage $u_{i\oplus j}$ and remove stages u_i and u_j.

      Update *data* to add $y_{i\oplus j} = y_i + y_j$ and remove y_i and y_j.

      Update *priors* to add $\boldsymbol{\alpha}_{i\oplus j} = \boldsymbol{\alpha}_i + \boldsymbol{\alpha}_j$ and remove $\boldsymbol{\alpha}_i$ and $\boldsymbol{\alpha}_j$.

    end

  end

  else

    indicator $\leftarrow 0$

  end

end

return *stage, score*

Algorithm 2: wr-HAC algorithm

For each hyperset $H_i \in \boldsymbol{H}$, we propose running wr-HAC $R \times \#(H_i)$ times. Here $\#H_i$ is the number of elements in hyperset H_i and R is a choice based upon

available computational resources. This sets the number of iterations of wr-HAC as proportional to the number of elements in the hyperset. We propose this despite the fact that the number of possible stagings, and therefore the search space, rises much faster than the number of situations. This is because if we chose the number of iterations as proportional to the number of stagings, all runs would be focused on the largest hyperset due to the super-exponential growth of the Bell numbers.

This means running wr-HAC has quartic computational complexity; each run has cubic complexity (as with AHC) and then wr-HAC is run based on the number of elements in the hyperset.

4.5 The Falls Example

4.5.1 The Dataset

Here, we provide an example of BMA to demonstrate our proposed methodology and show its benefits on a non-stratified dataset. The extension of functionality of `cegpy` used to create this example is available on github[3]. To do this, we work through its application to a simulated falls dataset of 50,000 individuals aged over 65 [Shenvi et al., 2018]. This dataset was chosen as it is simulated data in which the data-generating CEG is known. The event tree is constructed from the following five florets:

1. X_A: Individual living situation and whether they have been assessed (Communal Assessed, Communal Not Assessed, Community Assessed, Community Not Assessed)

2. X_R: Level of risk from a fall (High Risk, Low Risk)

3. X_{T1}: If an individual has been referred and treated (Not Referred & Not Treated, Not Referred & Treated, Referred & Treated)

4. X_{T2}: If an individual has been treated (Not Referred & Not Treated, Not Referred & Treated)

5. X_F: If a fall happened or not (Fall, Don't Fall)

The event tree describing this unfolding of the events can be seen in Figure 4.8. This is a non-stratified event tree because the process can unfold in a variety of ways. For example, for individuals that are not assessed for their risk of falls, it does not make sense to consider the outcome of their assessment.

[3]https://github.com/peterrhysstrong/cegpy_BMA/tree/dev

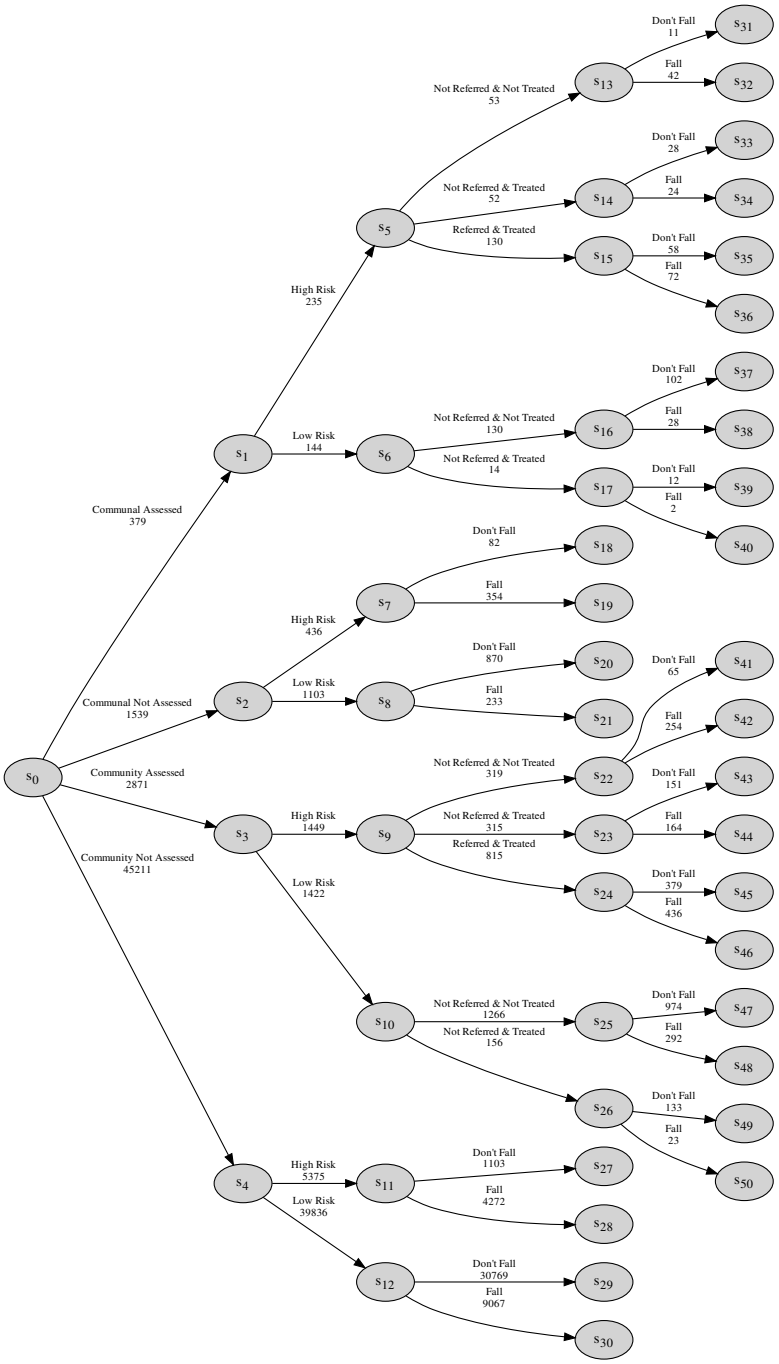


Figure 4.8: Event tree for the simulated Falls dataset with the counts for each path.

4.5.2 Input

For this example, we compare the results of running BMA on the full dataset compared to a random subset consisting of 10,000 individuals. This is to demonstrate

the impact that the sample size has on model uncertainty.

Here, we ran wr-HAC over each hyperset in the hyperstage. The five hypersets in the hyperstage, given in Equation (4.15), correspond to a hyperset for each type of event. For the other parameters we set $R = 100$, $\beta = 20$,$\gamma = 0.05$ and $\bar{\boldsymbol{\alpha}_0} = 4$.

$$\begin{aligned}\boldsymbol{H} =&\{\{s_0\},\{s_1,s_2,s_3,s_4\},\{s_5,s_9\},\{s_6,s_{10}\},\\ &\{s_7,s_8,s_{11},s_{12},s_{13},s_{14},s_{15},s_{16},s_{17},s_{22},s_{23},s_{24},s_{25},s_{26}\}\}\end{aligned} \tag{4.15}$$

4.5.3 Results

For both analyses– the full dataset and the subset– three of the five hypersets within the hyperstage have a single well-performing staging. These three hypersets share the same following unique well-performing staging: $\{s_0\},\{s_5,s_9\},\{s_6,s_{10}\}$.

Full Dataset

For the full dataset, hyperset $\{s_1,s_2,s_3,s_4\}$ also has a single well-performing staging, where each stage is a singleton. The remaining hyperset within the hyperstage– $\{s_7,s_8,s_{11},s_{12},s_{13},s_{14},s_{15},s_{16},s_{17},s_{22},s_{23},s_{24},s_{25},s_{26}\}$– has eight unique stagings outputted from wr-HAC; two of them are well-performing:

$$S_{5,1} = \{s_{11},s_{13},s_{22},s_7\},\{s_8,s_{12},s_{16},s_{25}\},\{s_{14},s_{15},s_{23},s_{24}\},\{s_{17},s_{26}\}\}$$

$$S_{5,2} = \{s_7,s_{11},s_{13},s_{22}\},\{s_8,s_{12},s_{16},s_{17},s_{25},s_{26}\},\{s_{14},s_{15},s_{23},s_{24}\}\}$$

.

As only one hyperset had more than one well-performing staging, the model average is over two CEGs which only differ in that one set in the hyperstage. The model weights for each well-performing model are shown in Figure 4.9.

The coarsest intersection of the stage which has two well-performing stagings is the same as the first staging, $S_{5,1}$, as the second staging is a refinement of the first. Both of the well-performing models are in a radius of one from their coarsest intersection. Therefore, the model uncertainty is around s_{17} and s_{26}: either both are in the stage $\{s_8,s_{12},s_{16},s_{25}\}$ or they are in their own stage.

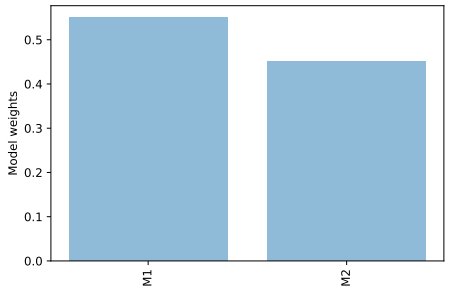


Figure 4.9: The two well-performing models for the full dataset with model weights given by the ratio of Normalised BFs

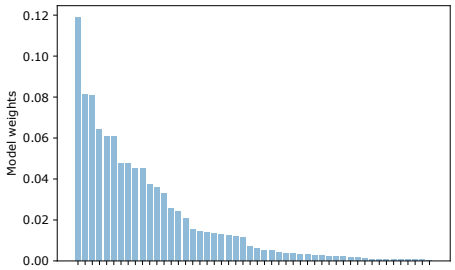


Figure 4.10: The 50 well-performing models for the subset with model weights given by the ratio of Normalised BFs

Subset of the Full Dataset

For the analysis on the subset of the data, there are two hypersets with more than one well-performing staging. Hyperset $\{s_1, s_2, s_3, s_4\}$ has 2 well-performing stagings which can be seen in Figure 4.11. These stagings are $\{\{s_1\}, \{s_2\}, \{s_3\}, \{s_4\}\}$ and $\{\{s_1, s_3\}, \{s_2\}, \{s_4\}\}$. Therefore, as one of these is a coarsening of the other, the coarsest intersection of these well-performing stages is $\{\{s_1\}, \{s_2\}, \{s_3\}, \{s_4\}\}$. This is the well-performing staging with the highest posterior probability; the other well performing staging is within a radius one of it.

Hyperset $\{s_7, s_8, s_{11}, s_{12}, s_{13}, s_{14}, s_{15}, s_{16}, s_{17}, s_{22}, s_{23}, s_{24}, s_{25}, s_{26}\}$ has 25 well-performing stagings, which can be seen in Figure 4.12. The coarsest intersection of these is given by

$$\{\{s_8, s_{12}, s_{25}, s_{26}\}, \{s_7, s_{11}, s_{22}\}, \{s_{23}, s_{24}\}, \{s_{13}\}, \{s_{14}\}, \{s_{15}\}, \{s_{16}\}, \{s_{17}\}\}.$$

All of the well-performing stagings are within a radius of five from this intersection. Therefore, the complete Occam's window BMA contains 50 well-performing CEGs, with their posterior probabilities (model weights) shown in Figure 4.10.

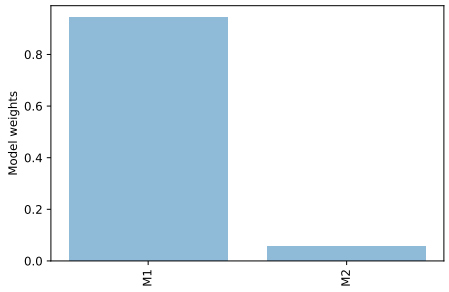


Figure 4.11: The two well-performing models for the 2nd hyperset of Equation (4.15) for the subset, with model weights given by the ratio of Normalised BFs

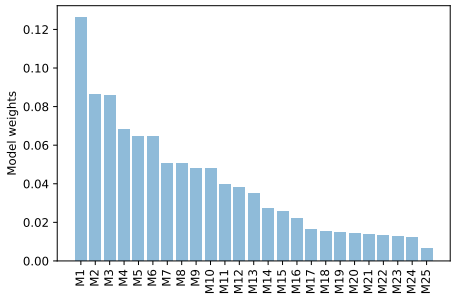


Figure 4.12: The 25 well-performing models for the 5th hyperset of Equation (4.15) for the subset with model weights given by the ratio of Normalised BFs

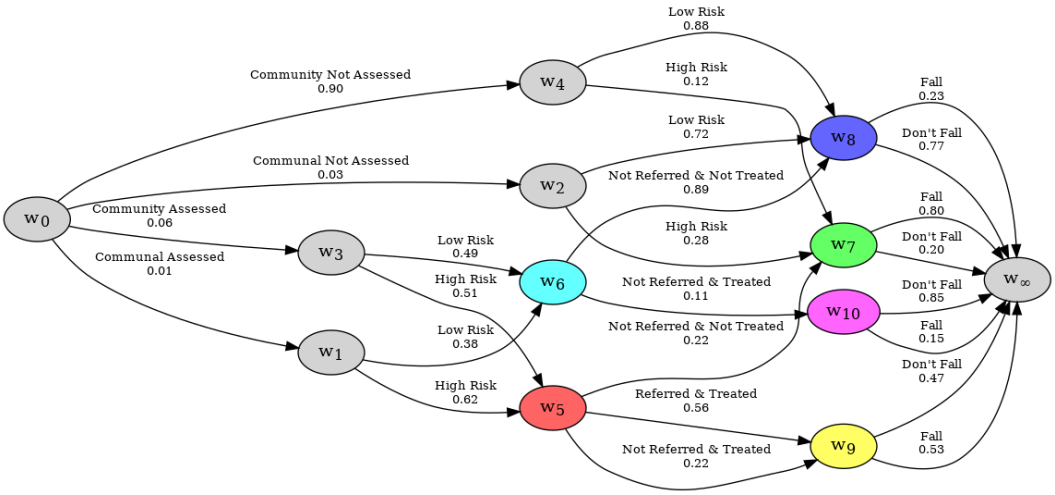


Figure 4.13: The CEG obtained via wr-HAC with the highest posterior probability for the full falls dataset with the mean transition probabilities given along each edge.

Comparison to AHC

For both the dataset and the subset, we also ran AHC to obtain the MAP estimate. In both of these settings, the MAP estimate obtained was the same as the highest-weighted model from the output of wr-HAC. However, these MAP estimates are understandably different from each other, as they relate to different datasets. The MAP estimates for the full dataset and the subset can be seen in Figure 4.13 and Figure 4.14 respectively.

The AHC MAP estimate of the full dataset gives the data-generating process (Figure 4.13). However, when using only the subset of the data, the MAP estimate

does not recover the data generating model. Note that, in the subset analysis, the coarsest intersection of well-performing models contains stagings which are not subsets of the staging of the data-generating process.

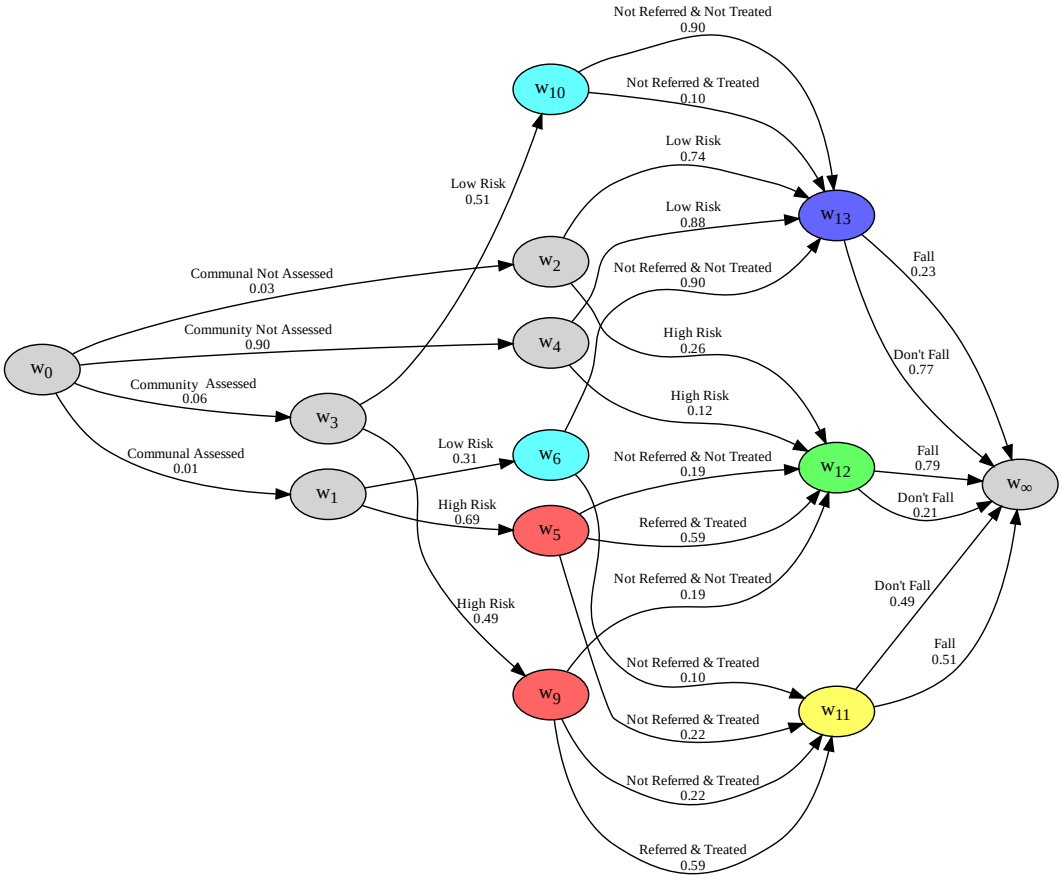


Figure 4.14: The CEG obtained via wr-HAC with the highest posterior probability for the subset of the falls dataset with the mean transition probabilities given along each edge.

4.5.4 Explanation of Results

In both the full dataset and the subset, wr-HAC provides only one well-performing staging in three of the five hypersets. These three stagings align with the staging in the data-generating process. Two of these hypersets have non-trivial stagings. For these hypersets, the unique, well-performing staging represents the following independence statements, which exist in all of the well-performing models:

- The outcome of the assessment for high risk individuals that were assessed is independent of living situation.

- The outcome of the assessment for low risk individuals that were assessed is independent of living situation.

Full Dataset

In the analysis of the full dataset, there is an additional hyperset with a single well-performing staging. This hyperset's staging represents the independence statement:

- The level of fall risk is not independent of living situations and assessment status.

For the only hyperset with multiple well-performing stagings, two well-performing stagings exist. The coarsest intersection of these stagings, as shown in Figure 4.13, corresponds to the following independence statements:

- Dark blue- If an individual is at low risk and has not been treated, their fall risk is independent of whether they have been assessed and where they live.
- Green- Other than the assessed individuals who have been treated, if an individual is at high risk, fall risk is independent of living situation and assessment.
- Pink- If an individual has been assessed as low risk and has been treated, their fall risk is independent of where they live.
- Yellow- If an individual has been assessed as high risk and treated, their fall risk is independent of where they live.

Subset of The Full Dataset

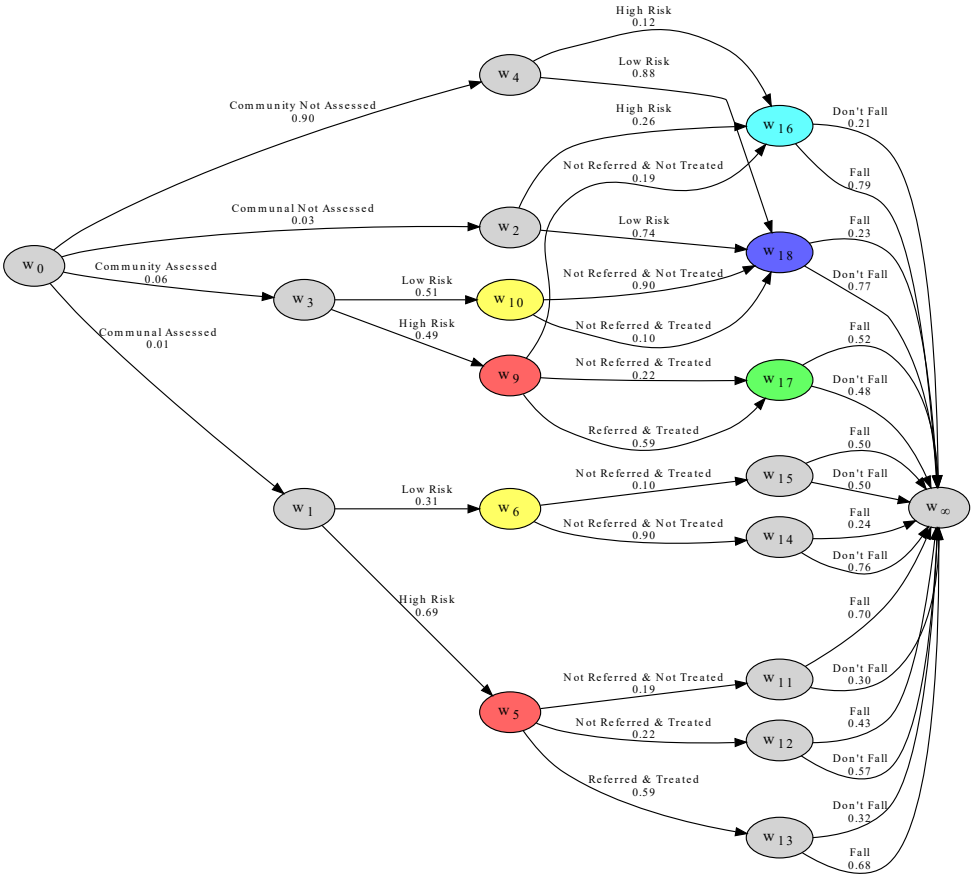


Figure 4.15: The CEG given by the staging of the coarsest intersection for the subset of the falls dataset with the mean transition probabilities given along each edge.

There are two hypersets in the subset analysis with multiple well-performing stagings: the 2nd and the 5th. In the 2nd hyperset, there are two well-performing stagings: $\{\{s_1\},\{s_2\},\{s_3\},\{s_4\}\}$ and $\{\{s_1,s_3\},\{s_2\},\{s_4\}\}$. The first corresponds to fall risk being independent of living situation and assessment and the second is that the level of risk in both the assessed populations– community and communal– are the same. As shown in Figure 4.11, the first staging, the more complex relationship, has a much higher weighting than the second. Therefore, there is much more evidence supporting the first staging.

In the 5th hyperset, 25 well-performing stagings exist. Therefore, there are many possible explanations that are consistent with the data. The non-singletons in

the coarsest intersection of well-performing stagings, as seen in Figure 4.15, correspond to independence relationships that exist in all of the well-performing models. The singletons in the well-performing intersection are the situations associated with Communal Living who were assessed. However, these situations were not singletons in any of the well-performing models.This suggests that, rather than these situations having a distribution over their edges different to any other situation in the hyperset, there is significant uncertainty about how these situations should be staged. As there are only 75 counts for Communally living assessed individuals and a weakly informative prior was chosen, it is to be expected that we have little certainty about the staging of these situations.

The non-singletons in the coarsest intersection of the well-performing models represent the following independence statements:

- Green- The fall risk is independent of whether they have been referred, if they have been assessed in the community as high risk and treated as a result.
- Dark blue- Other than the assessed individuals who lived communally, all low risk individuals have the same fall risk.
- Light blue- Other than the assessed individuals who lived communally, all high risk individuals who were not treated have the same fall risk.

For this dataset, the most refined union of the well-performing stagings is given by the CEG in Figure 4.16. Note that for the 5th hyperset, whether a fall happened or not, all situations are in the same stage. Therefore, there is insufficient data to determine any situations that should definitely not be in the same stage.

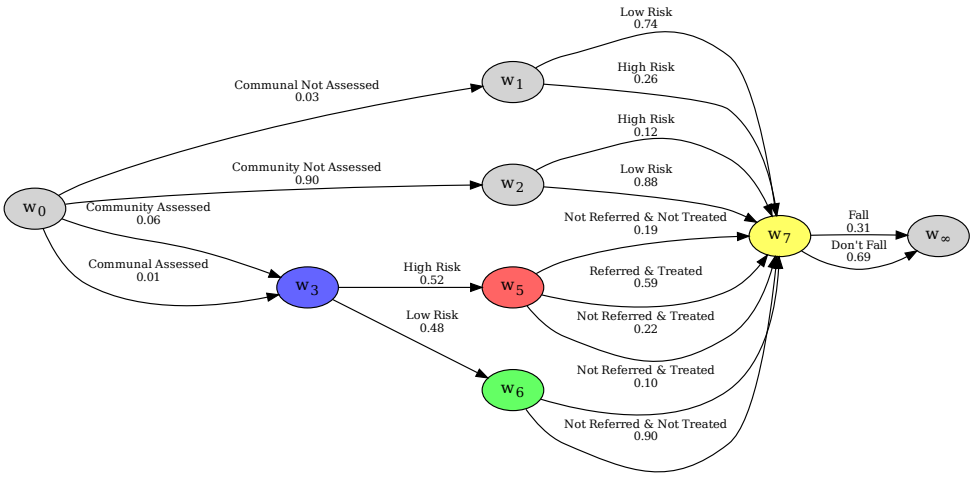


Figure 4.16: The CEG given by the staging of the most refined union for the subset of the falls dataset with the mean transition probabilities given along each edge.

4.6 Discussion

In this chapter, we have provided a methodology that enables a wider class of models, capable of dealing with asymmetric processes, to obtain the benefits of BMA. This class of models can model situations where using BNs would be wholly unsuitable, which significantly increases the applicability of BMA. The benefits of applying BMA for CEGs, by using a simple sampling algorithm, are clear: through exploring multiple well-performing models, the robustness of complex independence statements can be quantified by considering them within the set of well-performing models; when multiple well-performing models exist, the explanations shared by them can be extracted. Our choice of a naive sampling algorithm, wr-HAC, allows for model averaging in polynomial instead of super-exponential time. These are significant benefits compared to modelling that involves a single MAP estimate, where there is no quantification of the uncertainty of each independence statement. Although this could be done through diagnostic testing, the BMA approach provides an intuitive solution when diagnostic tests identify many models with high posterior probability.

Of course, although our naive sampler was sufficient to demonstrate the efficacy of model averaging methods when extracting robust inferences within classes of explainable models, it would be worthwhile to explore more sophisticated methods for extracting high-scoring models. Many alternative sampling algorithms are likely

to outperform wr-HAC and should ideally satisfy various attractive properties, such as the ability to scale up to problems with a much larger number of situations and approaches with guarantees about the rate of convergence. The methodology described in this chapter is not specific to our choice of algorithm and is more broadly applicable to any sampling algorithm. This further motivates the need for efficient model selection algorithms for CEGs. Newly available code – the `cegpy` python package [4] and the stagedtree package in R (for stratified CEGs) [Carli et al., 2020] – means that the practical efficacy of these can be more easily explored. We note that Markov Chain Monte Carlo model selection methods [Richardson and Green, 1997], such as those developed for mixture models [Kaplan and Lee, 2018] and for BNs [Madigan et al., 1995], are a promising avenue for further research into this area, for which the naive approach here can act as a baseline.

[4] https://pypi.org/project/cegpy/

Chapter 5

Scalable Model Selection for Chain Event Graphs: Mean-posterior Clustering and Binary Trees

Chapter 4 demonstrated how BMA can improve the robustness of inferences. However, there is still considerable need for efficient model selection algorithms for CEGs. As stated in Chapter 3, the most commonly used algorithm for model selection is the AHC algorithm, which scales cubically with the number of situations. This makes AHC quickly infeasible for all but the smallest of problems.

In this chapter, we define a novel, constraint-based approach for model selection for CEGs that scales quadratically with the number of situations. We begin this chapter by further motivating the need for faster model selection algorithms. Next, we define the totally-ordered hyperstage as a way of enforcing a constraint on the model space. We then explore the use of different functions to obtain the totally-ordered hyperstage. Next, we define binary CEGs (BCEGs), the transformation from a CEG to a BCEG and detail the benefits of such a transformation. We then compare the effects of using mean-posterior probabilities to give a totally-ordered hyperstage with AHC. The material in this chapter is based on Strong and Smith [2022a].

5.1 Introduction

Despite using a greedy search algorithm such as AHC (or w-AHC), which combines stages until there are no possible local improvements, as discussed in Chapter 4, there is still a need for efficient model selection algorithms for CEGs. To demonstrate this, suppose we had a hyperset with K situations, for which we need to find the optimal staging. We assume that $K > 1$. For the first iteration of AHC, there are $\binom{K}{2}$ possible stagings to consider. In the second iteration, there are $\binom{K-1}{2}$. In the worst possible case, combining continues until all situations are in the same stage. There are a maximum of $\prod_{i=0}^{K-2} \binom{K-i}{2} = \frac{K^3-K}{6}$ possible stagings that are considered. This gives a cubic time complexity on the number of situations [Nielsen, 2016].

As all possible stage combinations are examined, there may be numerous stagings which are insufficiently accurate to represent the underlying process. Therefore, this chapter aims to develop a model selection algorithm for CEGs that scales effectively as the number of situations increases. Therefore, one of our objectives is to eliminate inadequate stagings from consideration, retaining those that reflect the fundamental process well.

A second motivating factor was demonstrated in Collazo and Smith [2016]. When comparing stages that have very different effective sample sizes, strange optimal combinations can occur. This results in stages with large effective sample size tending to attract stages with a much lower effective sample size. More concerningly though, this occurs regardless of how far away these stages are in the probability space.

This problem is exacerbated by the sequence of pairwise steps that occur in algorithms like AHC. Once there is a combination of stages with larger and much smaller effective sample sizes, the combined stage has an even larger effective sample size, making it more likely to combine with other stages with small effective sample size. This behaviour is not ideal: it can lead to spurious staging. Therefore, as we are interested in reducing the number of potential stages that we consider in our model selection approach, we aim for our approach to prevent this spurious staging. Whilst this is a problem with the score function, a better-scoring model can be more spurious. We propose an alternative solution than to use a different score function such as one that uses non-local priors as in Collazo and Smith [2016], as these are more computationally expensive to calculate.

Here, we propose an alternative: a novel, heuristic search algorithm that reduces the set of models considered *a posteriori* by restricting the hyperstage. This prevents spurious stagings and leads to faster model selection than in existing

approaches.

5.2 Model Selection for CEGs

Model selection for CEGs is the task of exploring different partitions of the set of situations to find the MAP model. However, the number of partitions of a set expands super–exponentially on the number of the situations in the tree, as described in Section 3.4.3.

Therefore, dynamic programming approaches [Cowell and Smith, 2014; Silander and Leong, 2013], which consider all possible partitions of each hyperset, then become unfeasible for all but the smallest of problems. Greedy-search algorithms, such as algorithmic hierarchical clustering (AHC), are hence typically used (see Section 3.4.4). However, as noted in the previous section, AHC still scales poorly and it has been noted by multiple authors [Silander and Leong, 2013; Cowell and Smith, 2014; Strong and Smith, 2022b] that heuristic search methods are needed for model selection when there is a large numbers of variables. Model selection based on k-means for CEGs has been used but outputs worse MAP estimates than AHC [Silander and Leong, 2013]. Several score- and clustering-based approaches have been implemented in Carli et al. [2020]. In this paper, the cluster-based approach that combines situations if the distance between them – for various metrics – is less than a certain value performed comparably to AHC and is an interesting avenue of future research. However, these, like other clustering-based approaches, are not maximising the score function and therefore their score is highly dependent on the choice of clustering algorithm and hyperparameters used.

More recently, Shenvi and Liverani [2022] proposed using mixture modelling to perform model selection for CEGs that does not rely on conjugacy. This is critical when modelling holding times associated with the outcomes of events in dynamic CEGs where the distribution of the holding time may not be faithfully represented by a conjugate distribution. However, for the staging of situations, except when there is a trivially small number of stages, this approach takes longer than AHC.

Various papers have suggested *a priori* restricting the search space, for example by using a sub-class of CEGs such as cstrees [Duarte and Solus, 2021] or k-parent staged-trees [Leonelli and Varando, 2022]. *A priori* restrictions mean model selection becomes feasible and, because of their simplicity, their outputs are often easier to interpret. However, these approaches have two main drawbacks: firstly, restricting the model space *a priori* restricts the possible independence statements that can be represented; secondly, these sub-classes have no obvious extension to CEGs

which express asymmetric unfoldings of events [Shenvi et al., 2018].

5.3 Methods

5.3.1 Totally-ordered hyperstage

To describe our methodology, we introduce the concept of the totally-ordered hyperstage.

Definition 40 (Totally-ordered hyperstage) *Suppose an event tree T with a partitioning hyperstage, $\boldsymbol{H} = \{H_1, H_2, \ldots, H_N\}$, and a set of injective ordering functions, $f_n : H_n \to \mathbb{R}$, from each hyperset, $H_n = \{s_{1,n}, s_{2,n}, \ldots, s_{M,n}\}$. A totally-ordered hyperset is H_n with a strict total order induced by f_n, an ordering function, such that $s_{i,n} < s_{j,n}$ if $f_n(s_{i,n}) < f_n(s_{j,n})$. The totally-ordered hyperstage is a set of totally-ordered hypersets.*

Our model selection approach uses the totally-ordered hyperstage to restrict our model space. This is done by preventing stagings of non-consecutive situations in each totally-ordered hyperset. The totally-ordered hyperset is equivalent to the non-partitioning hyperset of each ordered set of pairs. That is $\{s_{i,n}, s_{j,n}\}$ is a hyperset if there is no $s_{k,n}$ such that $s_{i,n} < s_{k,n} < s_{j,n}$.

Example 41 (Totally-ordered hyperstage) *Here, we provide a toy example investigating the relationship between smoking and mortality to illustrate the use of the totally-ordered hyperstage. Suppose we have variables level of smoking (Never, Low, High) and whether the individual died during the period of the study. This is represented in the event tree shown in Figure 5.1. This has the hyperstage $\boldsymbol{H} = \{H_1, H_2\}$:.*

$$H_1 = \{s_0\} \tag{5.1}$$

$$H_2 = \{s_1, s_2, s_3\}. \tag{5.2}$$

Suppose that elicited expert judgement suggests a monotonic relationship between level of smoking and increased risk of mortality and any staging where this constraint was not satisfied is considered spurious. A spurious staging would be one in which those who smoked the most and those who had never smoked were in the same stage with the other situation, those who smoke an intermediate amount, not in this stage.

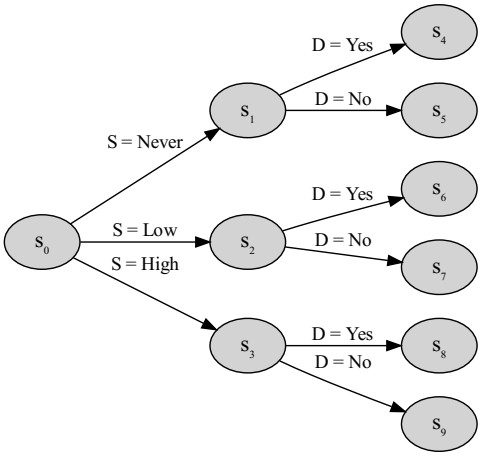


Figure 5.1: Event tree on smoking and mortality.

This can be represented in an ordering function $H_2 \to \mathbb{R}$*:* $f(S_i) \to i$*. This gives the ordering*

$$s_1 < s_2 < s_3$$

with the hyperset:

$$\{s_1, s_2\}, \{s_2, s_3\}.$$

The other hyperset H_1 *is trivially ordered as it is a singleton.*

The constraint used in this example is one performed a priori, such as the restrictions discussed in Subsection 5.2. A discussion on this sort of restriction using the hyperstage is given in Subsection 5.8.1.

The focus of this chapter is on constraining the model space *a-posteriori*. This makes it very different to existing methods as all stagings are possible before any data is collected. Using this method, none of the potential stagings of the CEG – and therefore relationships between variables – are ruled out before the CEG is fit to data.

5.3.2 Computational Complexity

This approach drastically reduces the size of the model space and therefore the time of a search algorithm. As discussed in Section 5.1 and repeated here for comparison, when running AHC on a hyperset with N situations, AHC would consider $\binom{N}{2}$ possible mergings in the first step. In the worst possible case, AHC would consider $\frac{N^3-N}{6}$ possible mergings in its run. In contrast, when running AHC on a totally-ordered hyperset with the same number of situations, it considers $N-1$ possible mergings

within the first instance, with at worst $\frac{N(N-1)}{2}$ possible mergings considered overall. Therefore, as N increases, the number of situations in the hyperset, the computational complexity of AHC on the totally-ordered hyperset grows quadratically, instead of cubically, improving its scalability as the number of variables increases.

5.4 Mean Posterior Probabilities

We propose using the mean posterior probability of each situation in the saturated CEG for each of our ordering functions f_n. This function only maps into $\mathbb{R}$ when the maximum number of outgoing edges from any situation is two, restricting this approach to binary trees.

Definition 40 also restricts our choice of ordering functions to those that are injective to allow a total ordering. Therefore, the choice of mean posterior probabilities is only suitable if these values are unique. To address this, where there are situations with the same mean transition probability, these situations are automatically placed in the same stage. For example, if multiple situations are given the same prior, they will automatically be placed in the same stage if there are no observations of that situation. We justify this choice as we wish for our search to be parsimonious: given the sample size, both observed and effective through the prior, what more evidence could there be that those situations should be in the same stage.

The mean posterior probability of a situation is the probability of each outcome at a situation. Therefore, choice of mean posterior probability is desirable as it introduces increased interpretability into the model selection process: if two situations are merged together, they must have a comparable probability of their outcomes relative to the set of situations in their hyperset. Note that, from a practical perspective, this prevents the AHC algorithm from combining stages spuriously [Collazo and Smith, 2016] where stages with a large sample size can absorb all stages with small sample size. Using the mean posterior probability as our ordering function, a stage with a large sample size can only absorb stages with small sample sizes if they have comparable, relative to the stages in the hyperset, mean posterior probability.

We define running the AHC algorithm using this choice of ordering function as the Mean Posterior Clustering (MPC) algorithm.

5.5 Binary CEGs (BCEGs)

As stated in Section 5.4, this approach only works on binary event trees – event trees where each situation has at most two outgoing edges. Therefore, to be able to apply this approach more generally we will consider the equivalence class of CEGs.

5.5.1 The Equivalence Class of CEGs

Research from Görgen and Smith [2018] and Görgen et al. [2022] proves how the statistical equivalence class of CEGs can be traversed through *swap* and *resize* operators and their inverses. Note that not all swap and resizes traverse the statistical equivalence class as some would lead to a loss of relationships between the distributions.

Swap

The definition of a swap is based on twins.

Definition 42 (Twin [Görgen and Smith, 2018]) *A twin around some stage u is the probability subtree $\mathcal{S}_u \subseteq \mathcal{S}$ where all root-to-subleaf paths have exactly two edges, and each child of the root is in the same stage u.*

The Swap Operator reorders the situations in a twin within the full tree. This means the trees are identical up to a change of ordering of the events with the same set of parameters.

Definition 43 (Swap [Görgen and Smith, 2018]) *Let $\mathcal{S}_u \subseteq \mathcal{S}$ be a twin. We define a map $\tau : \mathcal{S} \to \mathcal{S}'$ by exchanging the root of the subtree $\mathcal{S}_u$ with its children, creating a new subtree $\mathcal{S}'_u$. In $\mathcal{S}'_u$, the root will be in the same stage as the children in $\mathcal{S}_u$, and the children of the root of $\mathcal{S}'_u$ will be in the same stage as the root of $\mathcal{S}_u$. The tree $\mathcal{S}'$ is equivalent to $\mathcal{S}$, excluding the above subtrees, up to the vertical reordering of situations that lie downstream of the subtrees. The map τ is a* naïve swap *and is a* swap *if $\mathcal{S}'$ is a staged tree that preserves the stage structure and transition probabilities of $\mathcal{S}$ and which has an identification between their atoms.*

Example 44 (Swap) *An example of using a swap operator can be seen in Figures 5.2 and 5.3. In Figure 5.2, the situations are ordered such that Event A occurs before Event B; in Figure 5.3, the situations have been reordered– Event B occurs before Event A. Note that this preserves the independence statements.*

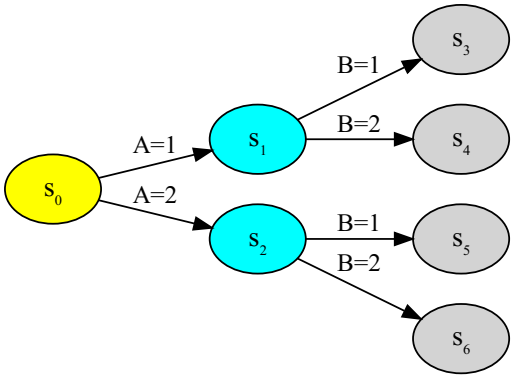


Figure 5.2: Staged tree where Event A occurs before Event B.

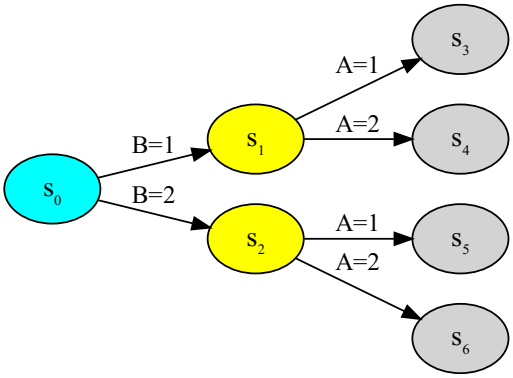


Figure 5.3: Staged tree after applying the swap operator: Event B occurs before A.

Resize

The second operator is the resize. The resize contracts subtrees in the event tree into a single floret while leaving the rest of the tree invariant, with each edge in the floret corresponding to a path in the sub-tree. The transition probabilities along the edges in the floret are the product of the transition probabilities on each of the root-to-leaf paths in the subtree.

Definition 45 (Resize [Görgen and Smith, 2018]) *Let $\mathcal{S}_w, \subseteq \mathcal{S}$ be a probability subtree. We define a map $\kappa : \mathcal{S} \to \mathcal{S}'$ that contracts $\mathcal{S}_w$ into a floret $\mathcal{F}_w$ with $\boldsymbol{\theta}_{\boldsymbol{\mathcal{F}_w}} = \{\pi_{\theta_{\mathcal{S}_w}}(\lambda) : \lambda \in \Lambda(\mathcal{S}_w)\}$, while leaving the rest of $\mathcal{S}$ invariant. We call κ and κ^{-1} a* naïve resize*, and a* resize *if $\mathcal{S}'$ is a staged tree.*

Example 46 (Resize) *Figure 5.5 shows the inverse of a resize applied to Figure 5.4. We have expanded the floret with Outcomes 1, 2 and 3 into a subtree with two florets: the first with outcomes 'Outcome 1' and 'Not Outcome 1'. The floret 'Not Outcome 1' leads to two leaves denoting the alternative outcomes, 'Outcome 2' and 'Outcome 3'.*

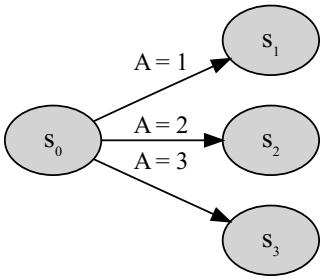


Figure 5.4: A floret with 3 outcomes

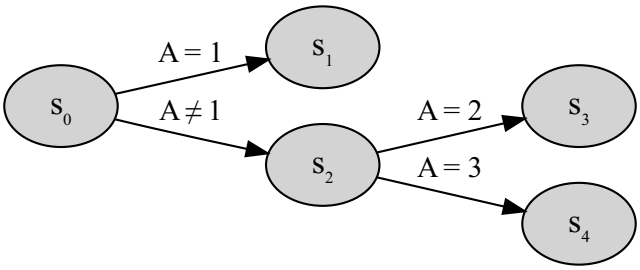


Figure 5.5: A floret where the inverse resize operator has been used

5.5.2 Binary Event Trees

The inverse of resize operators are of importance here, where a floret is expanded into a larger subtree. This describes an isomorphism from the set of originally considered CEGs to the set of BCEGs, with the same set of root-to-leaf paths. Therefore, each CEG has an equivalent representation as a BCEG: by first transforming a tree into a binary tree, we are simply embedding a search space into a bigger one. This means that any stage structure in the original tree can be preserved in the binary tree.

Representing any CEG as a BCEG has the benefit that more complex independence statements can be learnt. For example, when comparing situations in the same hyperstage with three outgoing edges, it is possible that multiple florets have the same distribution over two of their edges but not the third. This relationship could be captured in a binary tree but would be missed in the staging of the original tree.

As we are only considering square-free CEGs, the hyperstages of the binary florets created by the resizes are induced by the hyperstages in the original tree, with all other hyperstages remaining invariant.

Example 47 (Binary trees) *Here, we provide an example investigating the relationship between a person's sex and smoking habits to illustrate how binary trees can provide additional information. Suppose we have variables: sex (Male, Female) and smoking habits (Never, Low, High). This is represented in the event tree shown in Figure 5.6.*

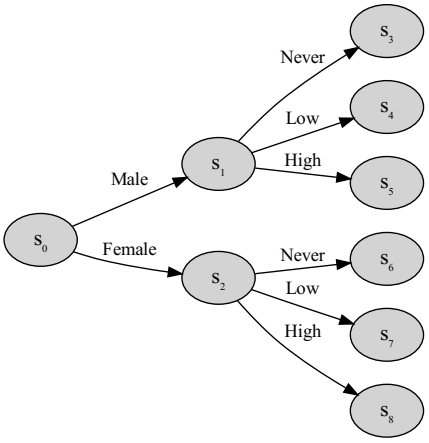


Figure 5.6: An event tree detailing a person's sex and smoking habits

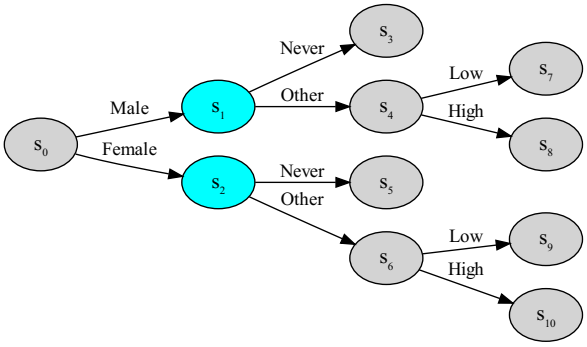


Figure 5.7: A staged tree detailing a person's sex and smoking habits after a inverse resize operator has been used

We create a binary tree by expanding the florets with outcomes: 'Never', 'Low' and 'High' into a subtree with two florets. The first has outcomes 'Never' and 'Other'. The outcome from the 'Other' floret leads to the outcomes: 'Low' and 'High'. The binary tree is shown in Figure 5.7.

The hyperstage of the original event tree is $\{\{s_0\}, \{s_1, s_2\}\}$ *with the hyperstage of the resized tree* $\{\{s_0\}, \{s_1, s_2\}, \{s_4, s_6\}\}$.

The BCEG is able to represent independence statements not possible in the original CEG. An example of one of these statements is shown in Figure 5.7: 'Never' smoking is independent of sex but smoking quantity is not. It is not possible to see this independence statement in the original tree due to its graphical representation. The resize from the tree in Figure 5.7 to the tree in Figure 5.6 is an example of a resize that does not traverse the statistical equivalence class as it does not preserve stage structure.

5.5.3 Computational Complexity of Binary Trees

To stage a binary stage tree, as there are more situations, more stagings need to take place. Suppose we have a hyperstage of a variable with N situations in it. When performing AHC, these N situations need to be staged. When performing MPC on the binary transformation of the tree, the N situations have to be staged $k_i - 1$ times, where k_i is the number of outgoing edges associated to that variable. When the original tree is binary, $k_i = 2$, we stage the same number of situations as AHC.

Therefore, the maximum number of considered stagings in MPC is given by

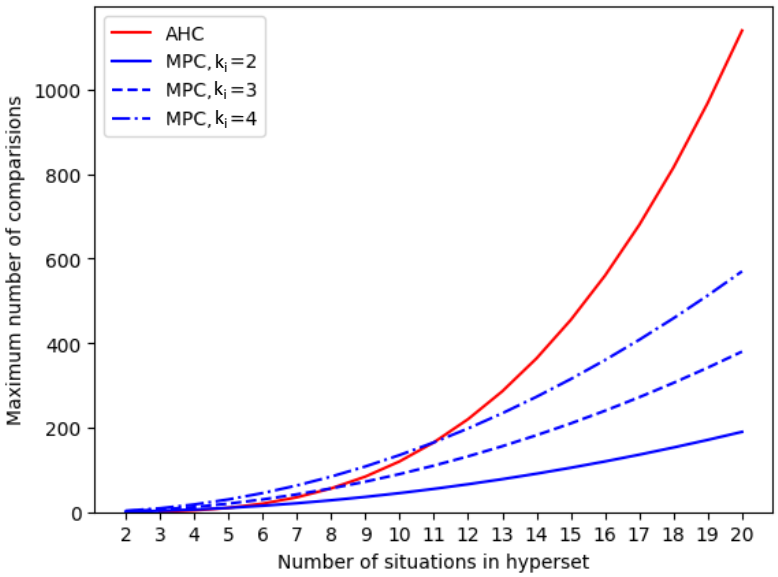


Figure 5.8: Maximum possible number of considered stagings for differnet model selection algorithms on different trees.

$(k_i - 1)\frac{N(N-1)}{2}$. Figure 5.8 shows how this, for different values of k_i, compares to the maximum number of considered stagings in AHC. This shows that for all but the smallest number of situations in the hyperstage, for which the total number of comparisons will be small, the number of comparisons in MPC will be significantly fewer than in AHC. This means that, although more stagings need to take place, MPC is still faster than applying AHC.

5.5.4 What Resize?

To represent the tree as binary, we need to decide what resize to perform. The choice of resize determines how the model space is expanded. This choice will be domain-specific, depending on the outcomes of interest. It is important to note that, regardless of the inverse resize chosen, all possible stagings in the original CEG have an equivalent staging in the BCEG. However, the extensions allow for the representation of different independence structures. For example, the independence statement represented in Figure 5.7 could not be represented in either of the other two possible resizes of the tree in Figure 5.6 with 'Low' or 'High' as the outcome of the first floret.

In this chapter, we focus our attention to inverse resizes that give a sub-tree for which every node is connected to a leaf. In this sub-tree, each floret will have two outcomes: one is an outcome from the original floret and the other represents

all other outcomes. This resize is given by an ordering on the edges, which decides the order in which florets give outcomes. Considering this type of resize with an original floret with k outgoing edges, there are $\frac{k!}{2}$ different resizes.

In the case where all potential resizes were considered, at most $k_i!(k_i - 1)\frac{N(N-1)}{4}$ comparisons would be made. This still only has quadratic computational complexity for the number of situations; therefore, for performing model selection when there is a high number of situations, MPC will still be faster whilst exploring a much larger model space. However, in this situation, care should be taken as certain stagings in the equivalence class will be represented multiple times.

5.5.5 Score Equivalence

To compare how well models represent the data, it is important that any two statistically equivalent models have the same score.

Recently, Hughes et al. [2022] defined a scoring function– BDepu– and proved it was score-equivalent. This was done by setting priors over the root-to-leaf paths and using mass conservation. This way of setting priors is invariant of the operators used to traverse the statistical equivalence class of CEGs. Therefore, we can set priors in a way that is consistent between non-binary and binary tree cases for fair comparison. For more details, see Section 3.3.1.

5.6 Comparative Analysis of Competing Methodologies

Here, we run comparisons of structural learning algorithms on a number of datasets chosen from existing literature on CEGs. These datasets are available from Carli et al. [2020]. We find the MAP estimate using MPC, AHC and AHC on the binary tree. The code used for this comparison was built as an extension on `cegpy` [Walley et al., 2022], a python package for learning CEGs. For each dataset, all processes were treated as stratified with any paths with zero counts added. We used the BDepu score to compare CEGs with different underlying event trees. For the purposes of this example, we constrain ourselves to using a fixed parameter, $\bar{\boldsymbol{\alpha}}_0$, as the number of leaves of each tree as done in Hughes et al. [2022]. The hyperstage was set so that situations relating to the same variable could be in the same stage.

The event trees were made binary arbitrarily using the following process: for florets with more than two outgoing edges selecting outcomes, by order of appearance in the data-frame, create two florets. One provides the selected outcome on one edge and the other edge leads to a floret with the rest of the outcomes. This process was

performed iteratively until the resulting tree was binary. For an example see Figure 5.10.

		Sample	MPC		Binary AHC		AHC	
Dataset	L	size	Time(s)	Score	Time(s)	Score	Time(s)	Score
Asym	16	1000	**0.04**	**-2423.67**	*	*	0.07	**-2423.67**
Pokemon	32	999	**0.21**	**-3251.94**	*	*	0.95	**-3251.94**
Titanic	32	2201	**0.17**	**-5235.56**	0.50	**-5235.56**	0.57	-5243.58
reinis	64	1841	**1.12**	-6715.61	*	*	11.5	**-6715.51**
PhDArticles	144	915	**9.41**	-4153.04	194.27	**-4152.25**	246.05	-4198.83
chestSim50000	256	50000	**70.46**	**-113458.66**	*	*	3336.28	-113458.87
monks1	864	432	**1137.25**	**-2663.73**	-	-	-	-

Table 5.1: Results showing the outcomes of the experiments. L represents number of leaves. Smallest time and largest score (BDepu) are in bold. An asterisks (*) is used to show when the original tree was binary. A hyphen (-) shows when the experiment timed out and took longer than 10,000 seconds. Experiments were performed on a laptop with 16GB of RAM with 4 core i7 2.6ghz cpu.

Table 5.1 shows the results of our comparison. MPC was the fastest model selection algorithm for all of the datasets considered, with it being orders of magnitude faster than AHC on the larger datasets. Regarding the BDepu score, the binary event trees had larger scores than the original event tree. MPC performed comparably to AHC. This illustrates that, although in the binary case it may consider far less partitions than AHC does, MPC can achieve similar, and often identical, scores than AHC in a much faster time.

When there are differences in the BDepu scores obtained by MPC and AHC on the binary tree, we can explore differences in the performances by comparing the stagings giving the score. Interestingly, we have observed that in some instances neither model gives the optimal staging, with a combination of the stagings of each model giving a better-performing staging. For example: there are two stages which are not merged together by MPC but are merged together in AHC; and one situation that belongs to one stage rather than another in AHC compared to MPC, with the model with the highest score not given by AHC or MPC but a combination of both their stagings.

It is also of interest to note that in Table 5.1, the time of AHC is faster in the binary tree than on the full tree when they are not the same. This is a surprise as, as mentioned in Section 5.5.3, there are more situations in this case and therefore more possible comparisons. Comparing stagings achieved by these models, this is not explained by the AHC on the binary tree stopping after performing fewer mergings. This suggests that some of the functions that are called as part of running

AHC are optimised to give increased performance to binary inputs. Therefore, this indicates, along with the results in Table 5.1, that doing model selection on a binary tree may be faster, even when using the same model selection algorithm, despite the increased size of the model space!

5.7 Christchurch Health and Development Study

This example uses a dataset from the Christchurch Health and Development Study (CHDS) as detailed in Barclay et al. [2013]. This study was conducted at the University of Otago, New Zealand [Fergusson et al., 1986] and is a longitudinal cohort study, taking place over 30 years of 1265 children born in mid-1977 in Christchurch, New Zealand. This dataset was chosen because an exhaustive search over all possible variable orderings has been performed in Cowell and Smith [2014] and therefore the MAP CEG is known.

As in Barclay et al. [2013] and Cowell and Smith [2014], we are interested in the following four discrete variables that relate to the first five years of the cohort for the 890 children for whom complete data was available:

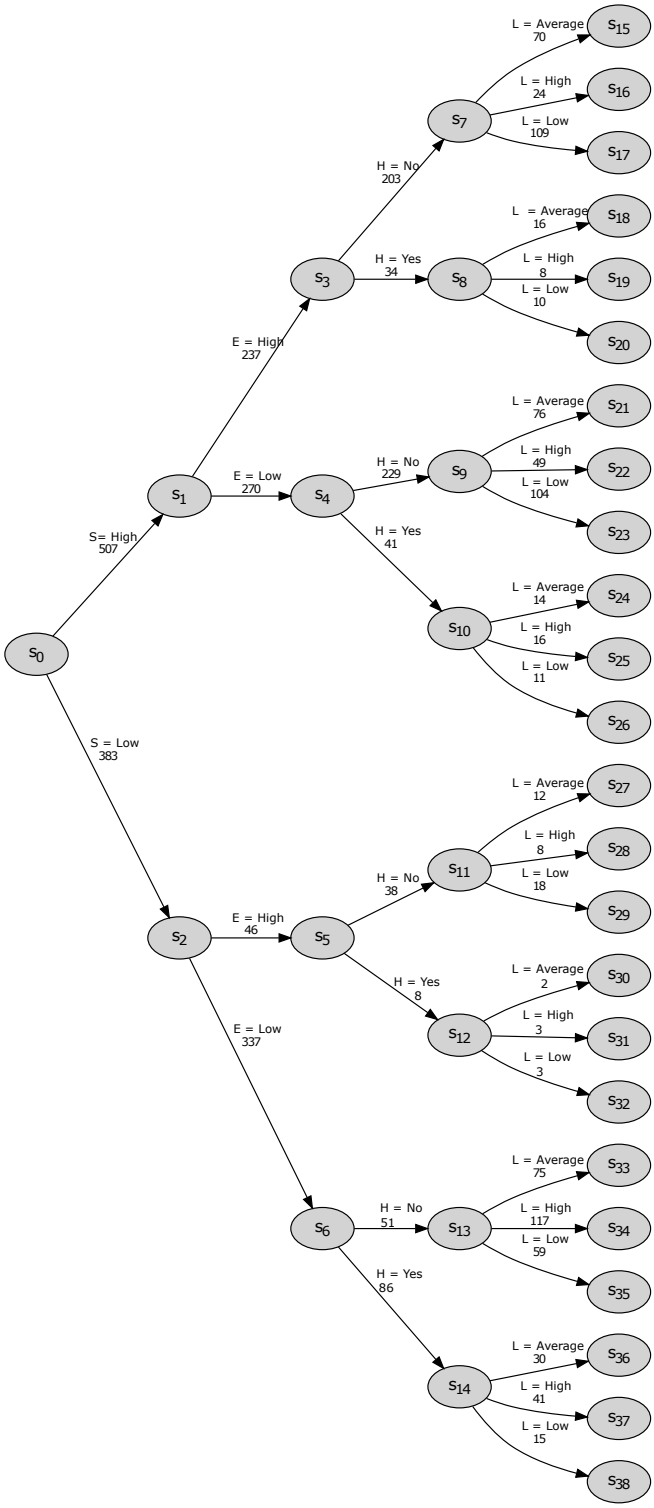


Figure 5.9: Event tree for the CHDS dataset with the counts for each path. S: social background; E: economic situation; H: admitted to hospital; L: number of life events.

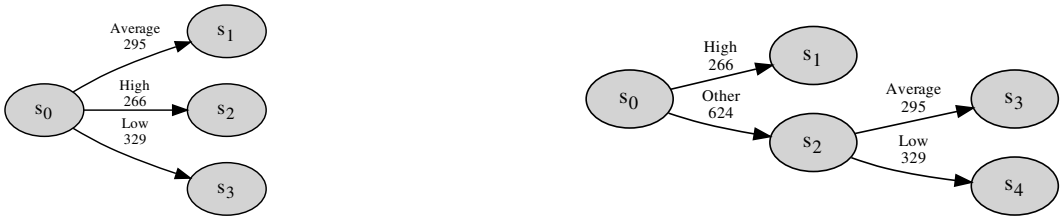
- X_S = Family social background– to henceforth be known as "social background"– categorised into High and Low levels based on maternal education, ethnicity, family social class and information concerning the child's birth.
- X_E = Family economic situation – henceforth to be known as "economic situation"– categorised into High and Low status dependant on income, accommodation, standard of living and financial difficulty.
- X_H = Child hospital admission, a binary variable accounting for hospitalisation during childhood.
- X_L = Family life events, classified as Low (0-5), Average (6-9) or High (10+) depending on the number of stressful events experienced e.g. death, unemployment or divorce.

Previous work on this dataset in Barclay et al. [2013] has shown how a CEG can outperform a BN on this dataset; Barclay et al. [2014] showed the above ordering is that which gives the highest-scoring CEG. For our methodology, we must first perform a inverse resize on the floret associated with number of family life events X_L to make the tree binary.

We split up the variable of life events into two binary florets:

- X_{L-high}: Was the number of life events High or not? (High, Other)
- $X_{L-average/low}$: If 'Other', was the number of life events Average or Low? (Average, Low).

This resizing is shown in Figures 5.10a and 5.10b. Using the binary florets gives the event tree in Figure 5.11. This embedding enables us to focus more closely on households that have a high number of life events.



(a) Floret X_L (b) Resize of X_L

Figure 5.10: Resizing of the floret X_L so that the event tree is binary.

The event tree for this dataset can be seen in Figure 5.9.

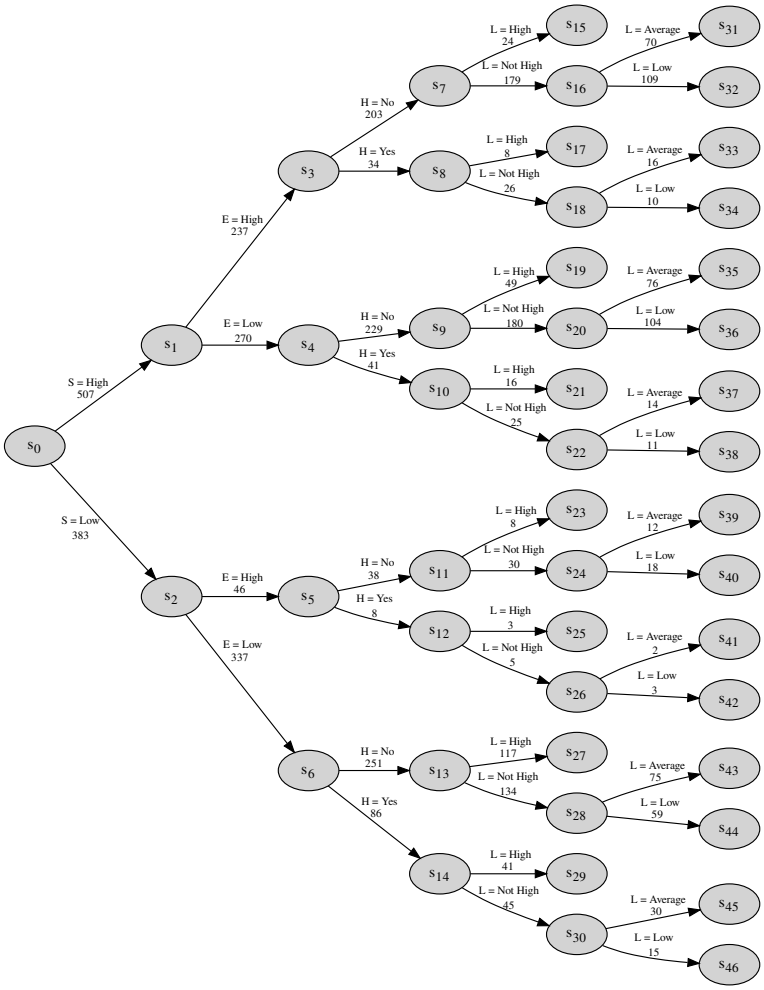


Figure 5.11: Binary event tree from the Christchurch Health and Development Study. S: social background; E: economic situation; H: admitted to hospital; L: number of life events.

The model selection in this example was set using the same strategy as described in Section 5.6. Running MPC on the binary tree gives the same output as running AHC on the binary tree. Both binary tree outputs outperform running AHC on the original tree. We also looked at the other two ways of making this tree binary – by resizing for average and low number of life events. Resizing for high numbers of life events gave the best BDepu score, although all three models outperformed the output obtained by AHC on the original tree.

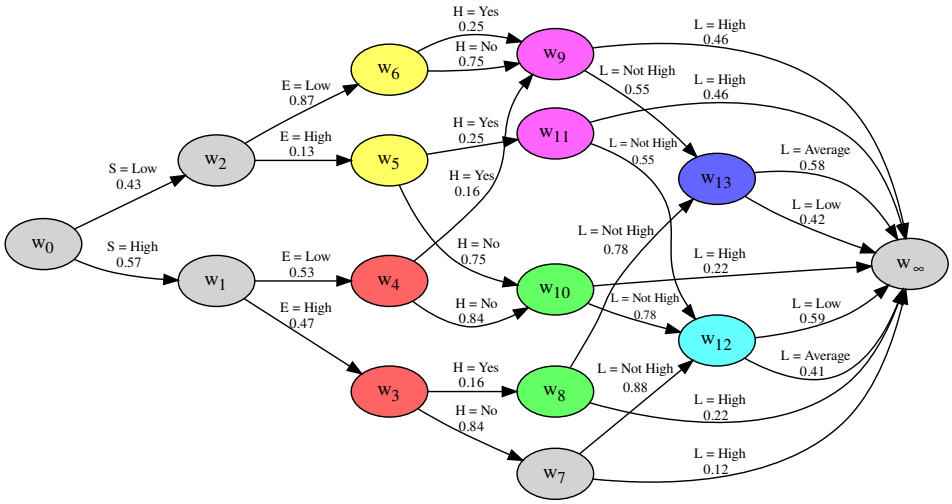


Figure 5.12: MAP CEG showing data from the Christchurch Health and Development Study. S: social background; E: economic situation; H: admitted to hospital; L: number of life events.

It is first important to note that for variables which are the same in the binary and non-binary trees – X_S, X_E and X_H – the staging is unsurprisingly the same, as the staging of this part of the CEG is unaffected by the resize.

However, for the part of the event tree that has been resized, X_L, new stage structure has been learned that is not present in the non-binary MAP CEG. Figure 5.12 has the situations corresponding to the variable X_{L-high} in three stages. These are ordered below with increasing probability of a high number of life events:

- There exists a single situation for individuals who have high social background and economic situation and were not admitted to hospital (w_7). These individuals have the smallest probability of a high number of life events.

- Individuals that have low social background, high economic situation and have not been admitted to hospital (w_{10}), individuals that have high social background, low economic situation and have not been admitted to hospital (w_{10}) and individuals with high social background and economic situation who have been admitted to hospital (w_8) have the same probability of high numbers of life events.

- Individuals that have low social background and economic situation (irrespective of hospital admission) (w_9), individuals who have high social background

but low economic situation and have been admitted to hospital (w_9) and individuals with low social background but high economic situation who have been admitted to hospital (w_{11}) all have the highest probability of a high number of life events.

This staging may appear complex at first glance but an intuitive rationale exists behind it. Our model shows that low social background, low economic situation and being admitted to hospital all compound the chance of having a high number of life events. Having none of these leads to a staging with the smallest mean posterior probability of a high number of life events. Being admitted to hospital (unless you have both high social status and economic situation), having low social status or low economic background are linked to a much increased probability of having a high number of life events; all of the other situations in this hyperset lead to a more moderate probability of having a high number of life events. The staging of $X_{L-average/low}$, where an individual does not have a high number of life events (w_{12}, w_{13}), does not follow the same pattern.

This shows that by expanding the model space, we can learn further insights from the data, which was not previously possible, as the staging obtained by MPC is not statistically equivalent to any staging on the non-binary tree.

5.8 Discussion

The MPC algorithm for model selection across CEGs is a novel structural search algorithm which outputs similarly – and often identically– scoring- models than the traditional AHC algorithm at a lower computational cost, because MPC scales quadratically rather than cubically. Through experiments and an example, we have illustrated the benefits MPC can provide for rapid computation and model accuracy. Note that the approach taken here is invariant of the choice of score function.

This work also further motivates the use of non-stratified CEGs. We have demonstrated, even for data that follows a product structure, we can obtain better scoring models by traversing the equivalence class into binary tree, which in many instances will be non-stratified.

We have also defined a new class of CEG: the BCEG. BCEGs provide increased explainability through expanding the model space, allowing for better-fitting models than non-binary CEG, irrespective of whether AHC or MPC is used. We have illustrated the benefits of BCEGs through experiments and an example, which identifies new independence statements not found in the non-binary tree.

The work in this chapter parallels work from Silander and Leong [2013] and

Cowell and Smith [2014], in which different orders of events are considered, corresponding to the swap operator. Here, we consider the space of models available to us using the resize operator. Both of these factors require thought in order to decide what model space to explore. This motivates the need for further investigation into the potential resizes that are used to make the tree binary.

5.8.1 Further Work

Constraining the hyperstage

Example 41 discussed using the totally-ordered hyperstage to represent an elicited monotonic relationship between variables instead of MPC. Here, we provide two natural extensions.

First, we could consider the relationship between a cyclic variable and the next staging. This has applications in time-related variables, for example days of the week or seasons, where we only wish to merge adjacent time steps together.

Example 48 (Cyclic variables) *Suppose we had the situations s_1, s_2, s_3, s_4 that represent the florets that are followed by the outcomes of each of the seasons of the year. The restriction of only wishing to merge adjacent time steps could be fulfilled using the following hypersets:*

$$\{s_1, s_2\}, \{s_2, s_3\}, \{s_3, s_4\}, \{s_4, s_1\}.$$

Secondly, we can consider the restrictions imposed by monotonic relationships of multiple previous events on future hypersets. Unfortunately, this constraint cannot be represented through a restriction on the hyperset.

Example 49 (Monotonic relationship restrictions) *We give an illustration of this process with the event tree represented in Figure 5.13. This event tree considers the impact of drinking alcohol and smoking on death. The smoking variable has three outcomes, with a monotonic relationship between them and mortality independent on other variables.*

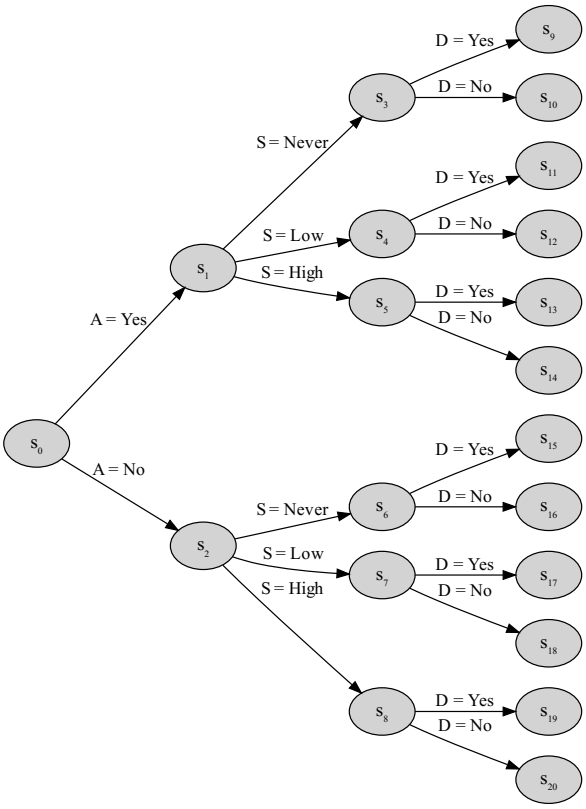


Figure 5.13: Event tree detailing impact of alcohol consumption and smoking on mortality. A = Alcohol consumption; S = Level of smoking; D = Death.

When considering the initial merging from one to two nodes, the totally-ordered hyperstage restricts s_3 and s_5 from merging; likewise, it restricts s_6 and s_8 from merging. All other combinations of pairs of situations give stagings that do not violate the monotonic restriction and therefore are hypersets. However, this hyperstage would allow the stage $\{s_3, s_7, s_5\}$ through successive combination of s_3 and s_7 and then this stage with s_5. This would create a staging that would break the monotonic restriction.

This shows that monotonic restrictions cannot be made through a constraint on the hypersets, as this restriction does not apply to stages locally and therefore would require a more sophisticated level of updating to determine what possible new stagings were allowed.

Note that, formally, such restrictions can be introduced directly into the score by setting a prior over the set of models, leading to low scores for any stagings that are not consistent with our constraint. This, however, would not lead to restrictions in how the model space is searched using these algorithms, just which models score

well.

Other ordering functions

Of course, there are various ways to embellish these algorithms further and we are currently investigating these, especially through the choice of ordering functions. Selecting the MPC over other ordering functions, such as Median or Mode, means that it directly relates to transition probability that would be given to a unit passing through. However, it would be interesting to explore the impact of considering other ordering functions in the future. The consideration of sample-means instead of mean-posterior as the ordering function was considered. However, this outputted worse BDepu scores than MPC on the experiments we ran.

Chapter 6

Chain Event Graphs of Agent-Based Models with Applications in Migration

Chapter 5 concluded that, even with a model selection algorithm that scales quadratically with the number of situations, model selection for CEGs with a high number of events is still infeasible for complex processes. In this chapter, we shift focus and consider how we can use elicited information in the form of an ABM and embellish it to create a CEG. Using expert elicitation and information already captured in an ABM can support model selection for CEGs. This chapter represents ongoing work, developed in collaboration with social scientists, the earliest part of which has appeared in Strong et al. [2022].

The chapter begins by providing a rationale for expert elicitation in Section 6.1. Section 6.2 explores the use of previously created models, specifically ABMs, as a form of elicitation to construct a CEG. An illustrative example of transforming an ABM into a CEG when the two models are similar is provided in Section 6.3, including details on the benefits of this approach. In Section 6.4, we explore how information can be elicited from an ABM into a CEG when the ABM contains features not naturally represented in a standard CEG, followed by considering how a wider class of ABMs can be embellished into CEGs in Section 6.5. To illustrate this, the chapter considers the migration domain in Section 6.6, where a model of migrant pathways is used as elicited information to create a CEG with desired properties. Finally, the chapter concludes by outlining exciting opportunities in this field and discussing plans for further developing this work.

N	# CEGs	# Root-to-leaf paths
1	1	2
2	2	4
3	30	8
4	124200	16
5	1.3016337×10^{15}	32
6	1.6669306×10^{41}	64
7	2.869365×10^{106}	128
8	3.219593×10^{264}	256

Table 6.1: Number of possible CEGs and root-to-leaf paths for a tree with N binary variables.

6.1 Motivation

We begin this chapter with a motivation of why expert elicitation is often needed to model processes.

When modelling complex processes with a large number of events, there are several problems that arise in CEG model selection. Firstly, as we noted in Section 3.4.3, the number of possible CEGs grows super-exponentially with the number of events. Further illustration of this is available in Table 6.1, which shows how the number of paths grows for a tree with binary events. As discussed in Chapter 5, there are several approaches that attempt to address this issue, including using greedy searches or restricting the set of models considered. However, as the model space increases, given there are no guarantees these methods find the model that best represents the data-generating process, we are less confident that the space is searched effectively.

Secondly and more pressingly, in larger models, the number of possible *root-to-leaf* paths, representing all possible unfolding of events, grows exponentially. For a tree with root-to-leaf paths of length N made up of events with k outcomes, there are k^N root-to-leaf paths. The first few values for a binary tree are shown in Table 6.1. For a binary tree constructed of 20 events, there are $1,048,576$ different root-to-leaf paths for a tree of this type. Therefore, even for large datasets, it is unlikely that every possible path will be represented, especially when the probability of certain events is low.

Further, as detailed in Chapter 4, even when all possible paths are represented in the dataset and the counts of each path are small, it is likely there will be many well-performing models, making inferences based on a single model overconfident. In summary, in complex model spaces, learning from data alone has several drawbacks.

This chapter presents a new methodology being developed to provide a Bayesian framework for ABMs. We apply this Bayesian framework to an existing ABM representing elicited information, transforming it into a CEG. There are many key benefits of this transformation detailed later in this chapter.

This work is ongoing and novel in that, to our knowledge, it is the first research that investigates how an ABM can be used to construct a CEG.

6.2 Agent-Based Models

ABMs are a computational model of a process, driven by the perspective of participants, known as 'agents'. They often involve simulating actions an agent undertakes in a given process, taking into account any relevant features and relationships inputted. Modellers must consider [Badham, 2020]:

- What types of agents are involved in the process being modelled.
- The characteristics of each agent e.g. their relationships with other agents, motivations, and any other information which may affect the outcome of the model.
- Any environmental features which may influence agent decisions.

ABMs are constructed by modelling the potential outcomes of successive events and decision-making [Epstein and Axtell, 1996]. To construct them, a range of data sources, such as large, structured demographic datasets or natural language narratives and theories are used to inform deterministic and stochastic transitions within an ABM. These transitions take the form of either mathematical equations, such as differential equations, or heuristic if-then rules and are informed by experts who describe the influences, possible options available and threats an agent might experience.

Due to their agent-centric perspective, interactions which dictate the behaviour of a complex system can be represented naturally through rules implemented in the model [Gilbert, 2019; Luke and Stamatakis, 2012]. As a result, they are increasingly used to model systems including crime [Groff et al., 2019], health [Tracy et al., 2018] and migration [McAlpine et al., 2020].

Despite their increasing popularity, ABMs are often unable to naturally combine expert judgement with available data to estimate and validate them. This is a particular problem in domains where it is difficult to gather large quantities of complete data.

Despite their ability to plausibly model the transitions of an agent, many ABMs have been described as opaque with many of the critical details needed to fully understand or replicate the models missing from publication. This is due to a lack of standardised model development [McAlpine et al., 2020; Hinkelmann et al., 2010; Grimm et al., 2006]. Some attempts, such as the ODD protocol, have been made to create a standardised structure for explaining ABMs [Grimm et al., 2006], but there is still significant variance in how the protocol is used and the clarity it brings to ABMs. ABMs' application often depends on the implementation of severely constraining software which may or may not match the modelled domain well. Perhaps even more concerning is the gulf that exists, when applying such models, between the domain and a principled statistical inference about that domain. In particular, no real guidance about how to set the ABM parameters is given; estimation of these is naive and model selection is performed simply by matching trajectories of hypothesised models with chosen/estimated parameters with sampled trajectories. As a result, others [Grimm et al., 2005; Heckbert et al., 2010; Schulze et al., 2017; An et al., 2021] have already identified the desperate need for embedding more principled ways of performing inference to estimate and validate ABM models when these are applied to real case studies. In this chapter, we argue that the best way of doing this is by using Bayesian models formulated around tree-based CEG methods in ways we illustrate below.

6.3 Benefits of embellishing an ABM into a CEG

6.3.1 Representation

As detailed in Section 3.2, a CEG is a compact representation of a staged tree, which represents the independence statements between events using colour. This makes it possible to visually identify any asymmetries in the sequence of events and the dependence structure that dictates the events transition probabilities explicitly from the graph's topology.

An ABM's typical representation – such as those in Figures 6.1 and 6.5– provides a flow diagram that shows the potential unfoldings of a series of events. This representation shows events by nodes and their outcomes with arrows, with multiple outcome arrows going to the same node if the structure of all future sequences of events are the same.

Example 50 (Agent-Based Model) *Some individuals are asked the same four questions with categorical answers, regardless of their previous responses. Each ques-*

tion is an 'event'. In a flow diagram, this process would be modelled by four nodes with all of the outcomes of Node 1 going into Node 2 and so on, with all outcomes of an event leading to the same next outcome. However, if all of the potential questions were different depending on the responses, then no two outcomes would lead to the same event. While an individuals' attributes and history may reflect the heterogeneity of their process, this is not easily unpicked or comprehended from model descriptions and visualisations in a flow diagram.

In contrast, the nodes in CEGs represent positions, events for which the potential future unfolding of events have the same distribution (and therefore structure). Therefore, a CEG is a natural embellishment of a flow diagram, which not only contains the sequence of events but also shows how the previous events impact their outcomes.

Example 51 (Agent Based Model continued) *Returning to the example, if the probability of an individual's answers are independent of the answers to the previous questions, then the CEG representation would be the same as that given by the flow diagram. More generally, whenever the probability of future events is independent of previous events, a Markov process, then the CEG representation will be the same as that of the flow diagram.*

One criticism of the CEG representation is that for sequences with many events, unless representing a simple dependence structure, few situations are in the same position. Therefore, the CEG is not that much more of a compact representation than a staged tree, with an exponentially growing set of situations with increased events. Here, we will detail two approaches that can be used, both individually and together, to mitigate this issue.

Firstly, when examining the CEG, it is possible to condition on certain events and therefore only examine parts of the entire structure at a time. This can be done using a probability propagation algorithm [Thwaites et al., 2008]. The techniques allows for focusing on circumstances of interest, making a potentially large and unreadable CEG interpretable.

Different representations of the CEG of the same equivalence class can also be used, as discussed in Section 5.5.1. There is currently no algorithm to reach the simplest CEG representation of the equivalence class but we can use resize operators to reduce the number of nodes in the graphical representation. For example, we can combine florets with a single possible outcome into the outcome of the previous event. The number of nodes in the representation can also be reduced by

a resize, when the dependence structure allows without removing any information as described in the example in Section 5.5.1. Due to the nature of the swap operator, the resulting CEG will still have the same number of root-to-leaf paths but traversing the equivalence class can reduce the number of positions, especially when examining a CEG conditioned on events of interest.

6.3.2 Bayesian Learning

As mentioned in Section 6.1, Bayesian methods are critical within complex processes because whenever models are sufficiently large to give a credible description of the processes, many parts are only sparsely observed. It is, therefore, critical to embed expert judgements through the use of priors, such as those on the hyperparameters. In this work, this is the distributions on the prior floret probabilities. In this way, our proposed methodology scales up to the granularity of descriptions shared by ABMs. We can embed not only the prior expectations of these probabilities – as often needed in typical ABMs – but also their uncertainty. This embellishment means that we can perform a prior-to-posterior update on these probabilities. In particular, we can derive principled model selection algorithms that respect the relative security of knowledge of different transitions within the system, through the strength of the priors. We note that, even if no actual steps in some of the paths are observed, we can proceed with this inference, whilst if many people are observed making a particular collection of transitions then estimated transition probabilities will be close to their sample proportions. The model is suitably regularised. Furthermore, if we assume floret independence, we can perform a conjugate Bayesian analysis (see Section 3.3). The consequent Bayesian model estimation and selection is both transparent and rapid due to the closed form representation and the interpretative understanding of the hyperparameters.

In particular, assuming each transition is multinomially distributed over the set of outcomes, to perform a conjugate analysis, we need to set the Dirichlet priors. The distributions for the transition probabilities are often not elicited in advance, due to the non-Bayesian nature of ABMs. However, if the values elicited are the mean transition probabilities, we can use these values as the prior means for the Dirichlet prior. In order to get the full prior distribution, we must add in a count of effective sample size. This acts as a measure of strength of the beliefs held within the ABM. This can be done either by eliciting such a value or by completing a sensitivity analysis around the value chosen, similar to the method taken in Shenvi and Smith [2019]. Other methods for setting up the hyperparameters can be seen in Collazo et al. [2018].

6.3.3 Model comparison

While developing a model, it is useful to compare it to alternative models which represent different hypotheses about the modelled process. Within a CEG, in order to compare competing models, we can set the hyperparameters so they match each other as closely as possible, as in Heckerman et al. [1995]. This is implemented via a mind experiment, where the strength of an expert's elicited opinion is expressed using phantom samples over potential root to leaf path developments. We then compare the marginal likelihood of different models, using Bayes factor to quantify the evidence supporting different hypotheses.

6.4 Eliciting a CEG from an ABM

In this section we define the class of ABMs that can very naturally be used to elicit a CEG and how this elicitation works. Within an ABM the heuristic if-then rules implicitly include independence hypotheses regarding the outcome of an event for an individual through the choice of inputs considered. By assuming these conditional independences within a hypothesised model, we can identify those agents within a sample who can be assumed on the next step of their journey to be exchangeable with each other. This is important if we wish to understand any processes through the relationships between unfolding events, and crucial if we wish to understand the impact of potential targeted interventions. The CEG provides a framework in which to embellish this model.

Of course, we could fit a CEG directly to model the migration process, through eliciting an event tree, the hypotheses and the prior distributions. However, if such an ABM has already been developed and thoughtfully calibrated to domain understanding – as is often the case – then it would be inefficient to ignore this information, even if it is only the starting place and the CEG later highlights areas for further expert elicitation. As we can exploit the fact that the CEG is largely compatible with the ABM, it can be used to embellish the original, rather coarse, description given by the ABM into an inferential model which is fit for purpose.

We note that other standard structural models such as BNs [Barclay et al., 2013], do not provide a good framework for egocentric modelling because the underlying processes and data tends to be highly asymmetrical and therefore does not allow a product space structure that is present in a BN. This is illustrated by the fact that ABMs – such as the ones used in the later application – typically need to use very different transitions depending on the current state the agent finds themselves in. BNs are also not able to represent context-specific independence

statements where an independence relationship holds only for certain values of the conditioning variable. The presence of context-specific independence statements is also common in this application; examples of such statements are provided in our illustrative example in Subsection 6.4.2.

6.4.1 Define the class

In order to use an ABM as elicited information to create a CEG, we must provide a proper formal, systematic description of an ABM – something that is sadly missing from many applications of this promising technology. Here, we follow Hinkelmann et al. [2010], who express the ABM as a particular class of dynamic system model where agents are variables and their transitions are given by local updating functions. This work provides a similar statistical framework through which to study ABMs. We consider a set of agents $(x_1, x_2, \ldots, x_n)$ that take values in $\mathbb{S}$ a finite discrete set that represents the possible states that an agent can be in. The set of all possible values of all of the agents in the system gives the state-space. For any given state in the space, the updating process that determines the transitions between states is a Markov process. The possible transitions in the Markov process can be represented by a directed graph $G = (V, E)$ with V the state space and edges $e \in E$ between $u \in V$ and $v \in V$ if it is possible to transition from state u to v.

To provide a comprehensive translation of general ABMs as formally described above into Bayesian stochastic models would be a massive task and beyond the scope of this thesis. Here, for simplicity, we constrain our attention to those ABMs with no agent to agent interactions, and with a Markov process that has graph representation in the form of a finite, rooted, directed tree. Later in this chapter, we discuss ongoing work considering the relationship between CEGs and a wider class of ABMs. For now though, the simplification of only using one agent is reasoned by the nature of these models being largely egocentric with the process and decision-making depending mostly on the state of the individual, even if affected by interactions with other agents and the environment. The rationale of only allowing a finite, rooted, directed tree for the updating of states is justified: we are interested in ABMs that can be thought of as an unfolding of a sequence of events. A tree gives the most natural representation of this process [Shafer, 1996].

This definition is justified as the type of information we have about single-agent process is best represented through an event tree representing the possible progress of each agent in a population. This is particularly useful as it depicts the step-by-step nature of the process, where each agent decides their next course of action, taking previous events into consideration. Typical hypotheses concerning

this progress assume various conditional independence hypotheses, such as those shown in Subsection 6.4.2. Within an event tree model, these can be expressed by the stage structure on the florets of the tree.

6.4.2 An Illustrative Example

Here, we introduce an illustrative example from Strong et al. [2022] of an ABM of an individual's decision on whether to migrate or not, represented in Figure 6.1. This decision is modelled as a sequence of events that impact their final decision. In this example, the ABM starts by initialising an individual's socio-economic status, X_I. The individual then may receive an offer to migrate, X_O. This offer either comes with or without employment, X_E. Finally, the individual makes a decision as to whether they should migrate or not, X_M. Each of the nodes in this diagram has an if-then rule associated with its transitions. For instance, Figure 6.1 shows an example heuristic rule for the decision to migrate. This rule shows how the probability of migrating is dependent on the outcomes of previous events.

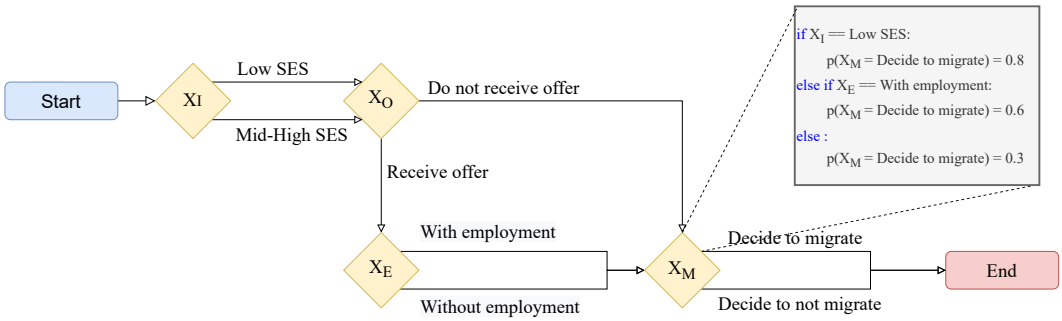


Figure 6.1: Example of an agent based model for migration. Here, 'SES' refers to socio-economic status.

As discussed in Subsection 6.4, by untangling the current representation, we can obtain an event tree which is implied by the ABM. Within this class of ABM, an agent's transitions are determined by the outcomes of their previous transitions. Therefore, the next transition is conditional on its previous events. Such events define the situations in the CEG, providing a direct link between the CEG and the ABM. The nodes in the ABM define the situations in the CEG, with the possible transitions from that node represented by the floret around that situation. The event tree thus obtained is shown in Figure 6.2. This is an example of an asymmetric unfolding of events; if the migrant does not receive an offer to migrate, we do not

need to consider whether the offer contains employment. This is denoted here as:

$$\nexists X_E | X_O = no. \tag{6.1}$$

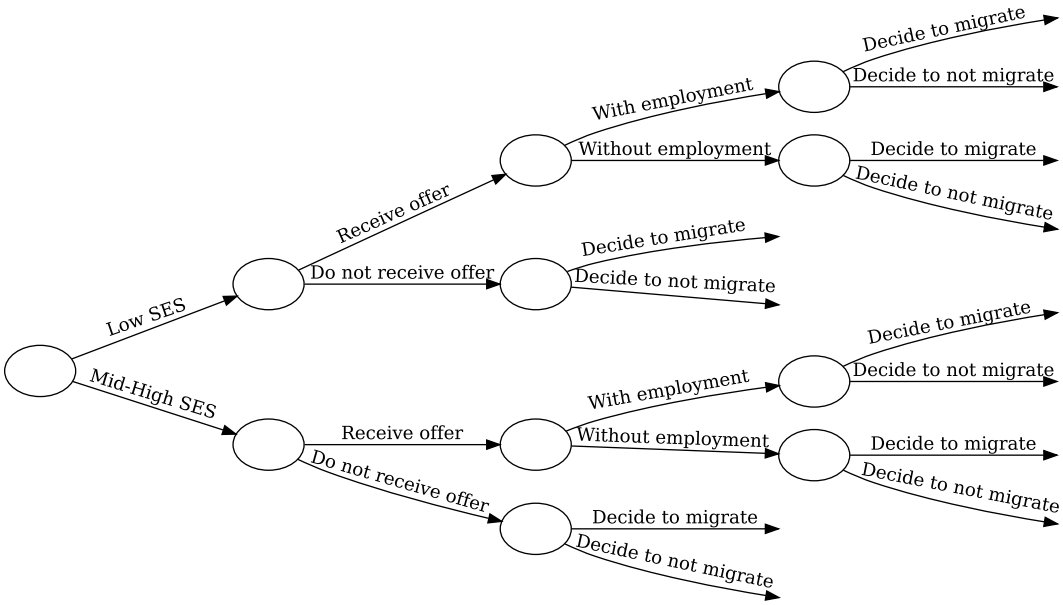


Figure 6.2: Event tree representation of the ABM shown in Figure 6.1. Here, 'SES' refers to socio-economic status. The leaf nodes are suppressed to prevent visual cluttering.

Next, by looking at the if-then rules within the ABM, we can identify the implicit independence statements that exist within these rules. For the decision rule regarding the decision to migrate, we have the independence statements:

$$X_M \perp\!\!\!\perp X_O, X_E | X_I = low \tag{6.2}$$

$$X_M \perp\!\!\!\perp X_O | \{X_I = mid\text{-}high, X_E \neq yes\}. \tag{6.3}$$

This provides the staging for the CEG. The staging can be represented by a staged tree, an event tree with florets in the same stage coloured the same. The staged tree for this example is shown in Figure 6.3.

For this example, we assume that the other rules in the ABM represent the following statements:

- W_2 (Yellow): Regardless of socio-economic status, the probability of receiving an offer is the same.

- W_3 (Green): When an offer is received, the probability of it containing an employment contract is the same, irrespective of socio-economic status.
- W_4 (Orange): A migrant with low socio-economic status has the same probability of deciding to migrate, irrespective of whether they have received an offer and whether their offer contained an employment contract.
- W_6 (Pink): A migrant with mid-high socio-economic status has the same probability of deciding to migrate if either (a) they receive an offer but it does not contain an employment contract or (b) they do not receive an offer in the first place.

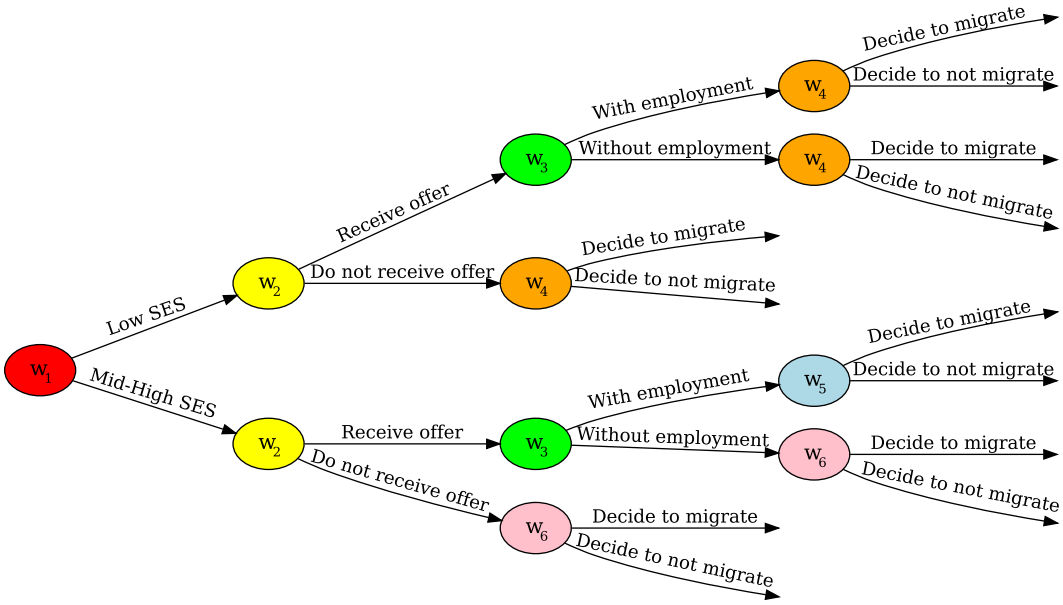


Figure 6.3: Staged tree representation of the ABM. Here, 'SES' refers to socio-economic status. The leaf nodes are suppressed to prevent visual cluttering.

From the staged tree, we can identify the nodes that are in the same position. In this example, w_4 and w_6 have the same future unfoldings for all future events, and are therefore in the same position.

Note that some nodes are the same stage but not the same position; w_3 is one such example, where the probability of the offer having employment is the same but the migrants' longer-term decision-making will still be influenced by their socio-economic status from earlier in the tree. This example demonstrates a context-specific independence statement: the decision to migrate is independent of whether you have an offer to migrate if your socio-economic status is low.

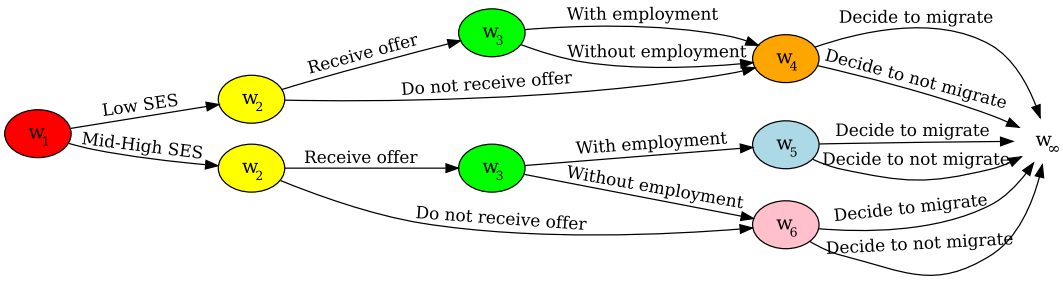


Figure 6.4: A CEG representation of the above ABM with some examples of independence statements. 'SES' stands for socio-economic status.

This example shows the CEG can model and provide a compact representation of the conditional independence hypotheses present in the ABM. The transformation from the ABM into the CEG now enables the natural transformation of the model into a Bayesian framework with its associated previously described benefits.

6.4.3 Causality

Often, ABMs are used for what-if analysis [Farmer and Foley, 2009; Gorman et al., 2006; Truszkowska et al., 2021]. These are used to give insight into different scenario outcomes such as what the results of different policy options could be [Badham, 2020]. This is a causal algebra: it makes explicit the– very strong– assumptions that take us from a model of an observational system into one that is manipulated.

This is comparable to a causal analysis done in a PGM. As CEGs are generalisations of discrete BNs, they share their accommodation of causal discovery algorithms. For CEGs, their own causal algebra has been developed [Thwaites et al., 2010].

It is important to not implicitly assume these properties. For example, assuming that a simulating model will continue to work after an intervention is made implies that the naive manipulation of the simulator parallels the manipulation made in practice. Secondly, it suggests that the real intervention will not have a knock-on effect on other parts of the system simulated by other components of the ABM.

To illustrate this, we use the example in Section 6.1. Consider the following:

- An intervention is implemented: it increases the number of offers being made with employment. This affects the number of migrants at stage w_3 being given an offer with employment.

- The simulated intervention may not represent the intervention in practice: as more offers include employment, they may not be as attractive as before as

benefits are diluted amongst the greater number of offers. This having the effect of decreasing the probability of those with offers with employment of migrating.

- This intervention may affect the wider model: before the intervention, all prospective migrants with low socio-economic status have the same probability of migration irrespective of being offered employment. If increased offers of employment were being made, migrants who had not been given these offers could be less likely to migrate; if they are aware of 'better offers', they may choose to 'hold out' in hope of something better.

Whilst this is a toy example, it illustrates why we should never automatically assume that a CEG, or any other model, chosen or selected for a system where no intervention is made can be extrapolated into a model that predicts what might happen under intervention.

6.5 Eliciting information from a wider class of ABMs

In this subsection, we explore whether a wider class of ABM than those previously described can be embellished into CEGs. This work is still ongoing and we outline exciting further extensions to the work already completed here to illustrate the potential of this application.

6.5.1 Scope of model

Firstly, it is important to state that some aspects of ABMs will not be suitable for embellishing into a CEG. Some ABMs focus on *emergent behaviour*: population-level behaviour caused by agent-to-agent interactions. *Emergent behaviour* is not self-evident from the programming of individual agents, but results from agents all interacting with the environment at once, influencing the decisions they make [Gilbert and Terna, 2000]. One example of an ABM to model emergent behaviour is in traffic modelling, where individual agents make decisions on where to drive but the overall emergent behaviour dictates where blockages and congestion occurs. Classes of models which are more suitable for embellishing with CEGs are those with a focus on understanding individual pathways and the dependence structure between them.

Therefore, in larger models of highly complex processes, some parts may be more suitable for embellishing into a CEG than others. An effective strategy for ensuring that this model is comprehensive and logical is to implement an Integrated

Decision Support System (IDSS) [Barons et al., 2018] by modularising the complex model into sub-models which interact with each other throughout the process. The IDSS provides opportunities for coherent inference throughout the network of sub-models, creating a comprehensive narrative through the multi-faceted probabilistic interdependent systems. Inference can be validated because it can be performed component by component, curated by relevant experts. This provides the opportunity for CEGs to be used where they are most appropriate, as a sub-model, and for other models – such as those related to demography or, indeed, ABMs– to be used at the points where they will be most effective. The IDSS structure maximises the efficacy of the entire system, acknowledging that a single class of models– or a single research domain– cannot best represent all parts of a complex process. The IDSS has so far mainly been used for dynamic BNs and Multiregresssion Dynamic Model classes but has also been recently used to model the interactions of several agents using dynamic CEGs supplemented by another component represented by an undirected graphical model [Shenvi et al., 2023].

Using an IDSS streamlines the model, meaning each sub-model only uses the information necessary for its accurate representation; it is parsimonious. By only including the parameters needed to explain the model effectively, we can be more confident of the inferences our model represents. Although low parsimony models (with more parameters) tend to have better fit, introducing many parameters can increase model uncertainty, confuse relationships between variables and become overly specific [Epstein, 1984].

In contrast, ABMs themselves have the potential to become unparsimonious very quickly. As O'Sullivan et al. [2012] describe, the increasing ease with which an ABM can be developed has led to a plethora of overly complex models, which attempt to explain systems but have become too complex to explain themselves : "we simply replace one difficult to understand phenomenon– the world itself – with an equally hard to understand model" (pg. 113). This highlights the importance that any additional features added to a model must make a difference that matters; although it is tempting for modellers to use additional parameters, purpose must be at the forefront of any additional complexities added into the model [Edmonds, 2017].

Whilst we are using the ABM as a representation of expert judgement, it is possible that artefacts could exist in an ABM [Galán et al., 2009]. An artefact is a mismatch between the set of assumptions in a model that the modeller believes is creating a phenomenon and what is actually creating the phenomenon. These mismatches can come in various forms such as using existing software or properties

of the domain which do not align with the assumptions of a type of model. These artefacts also exists within statistical models: for example, the choice of how something is distributed is often due to desirable properties of that family of distributions which may not faithfully represent the underlying process. Therefore, when mapping from an ABM to a CEG (or any translation from one model to another), it is important that the features being transferred represent the process being modelled rather than features of the existing model.

6.5.2 Events with continuous outcomes

One of the immediate differences between ABMs and CEGs is that ABMs often have continuous events whilst CEGs have only been defined on discrete spaces. When the continuous nature of the events is a key part of the process being modelled, this may be an example of a situation for which CEGs are unsuitable.

However, in many instances, discretising the model may be more appropriate and transparent than using continuous variables: when outcomes are treated as continuous, there are often implicit assumptions of their distributions, such as it being Gaussian, which may be inappropriate and therefore be even more constraining than discretisation. For example, if the outcome relates to an amount of money, which impacts whether future events take place, it may be instead more suitable to categorise individuals by the amount of money they have, splitting the space and probabilities of future actions they can take.

6.5.3 Dynamic processes and time

ABMs often contain recurrent events. As stated in Section 6.4.1, these ABMs cannot represented naturally in vanilla CEGs. However, these processes can be represented in a Dynamic-CEG, DCEG [Barclay et al., 2015]. This is a CEG on an infinitely large event tree, given by sequences of events which have no end.

Time is also often a part of an ABM's modelling process. Whilst by default a CEG has no time associated with the events outcomes, these can be incorporated thought the presence of holding times associated with their outgoing edges in a Continuous Time DCEG, CT-DCEG [Shenvi and Smith, 2020b]. In a CT-DCEG, each event has conditional transition probabilities and conditional holding time distributions (which may have a different dependence structure).

Time can also be an important part of ABMs where probabilities and dependence structures change over time, for example, when measuring academic performance across different exam seasons. Freeman and Smith [2011b] provide an

alternative Dynamic Chain Event Graph which models this sort of time series development, using a multivariate time series on the probabilities in the discretised model. Here, the staging of the CEG changes over time. Freeman and Smith [2011b] show how fast, closed-form model selection algorithms can be adapted from standard ones for discrete time modelling. These allow for both drifts on the probabilities and change points in the current staging of the process. This and other methods could be seamlessly transferred to this domain.

6.5.4 Multi-agent models

Most ABMs contain multiple agents (and some even contain multiple classes of agent types such as migrants and recruiters) and the interactions between these agents are a core component of the model. There are two established ways that these models can be embellished into a CEG. An important consideration of this process is defining a 'unit' in the model– in the case of migration, this could be an individual or family group. Parsimony must, again, be integral to this decision: where migrants are moving as a family, it would be more parsimonious to consider them as a single unit than as individuals.

The first and simplest approach is to directly model the interactions as a sequence of events happening to an individual. This differs from the way interactions in ABMs are often modelled, where agents' movements dictate their proximity– and therefore their interactions– with other agents. In formulating this process as a CEG, the probability of interaction is directly modelled instead. For example, in an ABM, it is common for agents' movements to be governed by random walks with interactions decided by proximity of other agents moving near them. This would give a probability of two agents interacting which could be modelled instead. If the spatial element of the model is not of key significance to the model, unlike in the traffic modelling example, then by modelling the interactions directly, a more parsimonious model can be used. In an ABM, interactions are modelled as actions between specific agents (e.g. between a specific migrant and a specific recruiter). In contrast, a CEG would model the probability of the different outcomes of interactions, treating them as interactions within a relevant sampled population (e.g. between any migrant at that stage in the CEG and any recruiter). Due to the nature of CEGs, the population from which the interactions are sampled from is dependent on previous events outcomes.

Secondly, another approach is to embellish the ABM into the framework provided by a CEG as part of a hierarchical model. This has been done for modelling terrorist groups with each individual in the group having a CEG within an IDSS

representing their interactions [Shenvi et al., 2023]. This demonstrates the potential to import ideas into an IDSS, making it possible to model agent interactions and outcomes jointly within the model class.

6.6 Application

6.6.1 Models of Migration

Rationale

Researchers and policymakers are interested in modelling migration as they aim to understand the mechanisms involved to inform policy. For example, organisations may aim to promote safe labour migration in line with the UN's Sustainable Development Goals to promote decent work, eradicate forced labour and end modern slavery and human trafficking [United Nations, 2021].

Migration, particularly low-wage labour migrants or migrants in conflict-affected settings, experience increased vulnerability to human trafficking and exploitation. It is estimated that 23% of victims of forced labour [International Labour Organisation, 2017] and 60% of victims of human trafficking were outside their country of residence [United Nations Office on Drugs and Crime]. In order to inform policymakers attempting to prevent exploitation, it is important to understand migrants' journeys and identify how individuals' hyper-precarity and livelihood insecurity, experienced due to both employment and immigration, evolves on different migration pathways [Lewis et al., 2015]. Migrants' pathways are often complex and non-linear, making many conventional modelling approaches unsuitable. The aim of these models is to accurately replicate a population, its environment and the interactions that occur.

History of migration modelling

Interest in modelling migration was initiated in the late 19th Century, when Ravenstein [1885] identified areas of 'absorption' and 'dispersion' in Britain using the 1881 Census. Ravenstein's "seven laws of migration" provide an insight into his analysis and the relationships he identified. Over the following decades, researchers began to compare the migratory behaviours of different demographic groups, initially beginning with sex differentials, but increasingly considering migrant status with respect to age, occupation, family status and motivation [Greenwood and Hunt, 2003]. In the 1940s, Stewart [1947] applied the gravity model to migration, in contrast to previous behaviourist ideas. Developments throughout the 20th Century included

developing the gravity model further and developing systemic migration modelling [Alonso, 1986; McFadden, 1981]. More recently, Agent Based Models have become increasingly popular; Section 6.6.2 provides further details on these models.

Challenges with modelling migration

In the study of demography, there are three key aspects of population dynamics: fertility, mortality and migration [Bijak, 2022]. Of these three processes, migration is that which has the most uncertainty and complexity [Council et al., 2000]. These issues are further compounded by different definitions of migration that exist in different organisations and the false dichotomy between voluntary and forced migration [Erdal and Oeppen, 2018].

Modelling any system requires large amounts of high-quality data. Although some data sources – such as censuses – can provide the information needed, many are incomplete or do not deal with more informal methods of migration and forced migration [McAlpine et al., 2020; Kraler and Reichel, 2011]. When studying migration pathways, as we are here, modellers are often reliant on migrant testimonies and the additional complexities which they bring. Often, data has small sample counts and high uncertainty; migrants may choose to not report all of their experiences to protect their interests. Furthermore, securing interviews with populations of marginalised, undocumented, or irregular migrants can be a huge challenge and, if secured, researchers face serious issues of time scarcity, language barriers, and availability to follow up [McAlpine, 2021]. Survey data can be used but is not applicable to entire migration models; rather, it can provide insights for individual transitions.

6.6.2 Agent Based Models of Migration

In recent years, ABMs have become increasingly popular for modelling migration [McAlpine et al., 2020; Entwisle et al., 2016; Fu and Hao, 2018] due to their ability to explore 'causal' complexities which are inherent to populations and human behaviour. Their 'bottom up' approach – focusing on individual-level decisions to form an aggregate macro-level behaviour– can provide deep insights into the system as a whole, with the aim of replicating its environment, populations and patterns so as to recreate the observed outcomes in the real world system.

McAlpine et al. [2020]'s review of ABMs in migration research identifies that current ABMs are overly-simplistic and the model development does not capture the range and diversity of factors that impact on migrants' behaviours and outcomes. Many of the models assume migrants are making rational choices to optimise out-

comes and do not reflect the true complexity of migrant decision-making processes. Sometimes, decision-making was simplified and did not include the core elements social scientists have previously identified as being crucial to migrant decision-making processes.

6.6.3 ABM of Thailand to Myanmar migration corridor

Section 6.4.2 illustrates an idealised example of when an ABM and a CEG are the same. In reality, due to the different methodology used to create the models, there is likely to be differences in these models due to the focus of the modelling technique being on different processes.

In this section, we introduce an ABM of the migration process between Myanmar and Thailand, taken from McAlpine [2021]. We then explore how aspects of this model can be used as elicited information to provide a CEG.

The Myanmar-Thailand Migration Planning and Intermediary Networks (MyTh MAP-IN) ABM explores the experiences of economic migrants moving from Myanmar to Thailand, both through regular and irregular migration pathways. It aims to provide a conceptual understanding of the relationship between choices migrants make and the precarity of their situation.

The MyTh MAP-IN ABM uses three agent classes: migrants, intermediaries and employers. Each class has different characteristics and properties. The model environment includes areas which represent the origin and destination locations and crossing points where migrants can move between locations.

The model consists of four sub-models, corresponding to parts of the migrant's journey (pre-migration, planning, transit and employment). Each sub-model requires agents to act differently according to the decisions required in that sub-model; some processes occur of multiple sub-models. Figure 6.5 shows the computational model and decisions which are made by agents over the ABM[1].

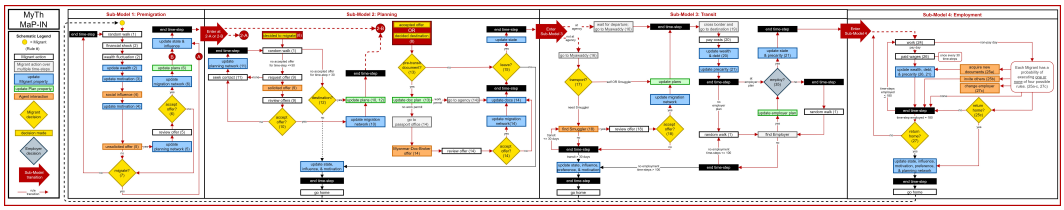


Figure 6.5: Computational model of MyTH MAP-IN ABM, taken from McAlpine [2021]

[1]Model code and documentation for the MyTh MAP-IN ABM can be found at: https://github.com/feature-creature/MyThMaP-IN

6.6.4 Embellishing into a CEG

Some elements of the ABM can easily and effectively be translated into a CEG. We will show how this works in practice using an example from the third sub-model MyTH MAP-IN ABM, describing how a migrant decides to move to a new country. The rules that describe the modelling of the decision are given in Figure 6.6.

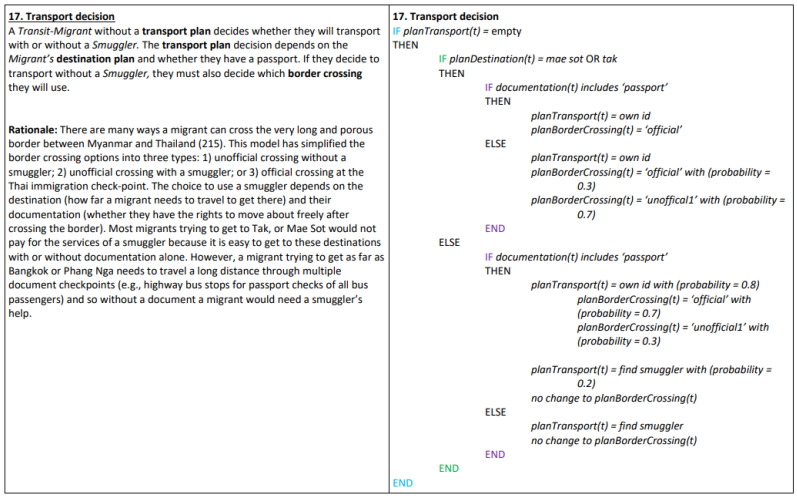


Figure 6.6: Part of the MyTh MAP-IN ABM, describing a decision about a border crossing. Taken from McAlpine [2021]

Example 52 (Border crossing example) *This decision is based on where their planned destination is and whether they are in possession of a passport. An individual decides to either cross the border by themselves or with a smuggler and, if by themselves, whether to go officially or unofficially. This state is conditional on them not having a plan to cross the border in advance, such as having transport organised through a recruiter.*

In this example, there are a finite number of variables of interest: presence of passport, destination location plan, crossing alone or with a smuggler, and whether they cross officially or not. Each of these variables has a finite number of discrete values; in this case each is binary: passport or not, location A or B (used as a proxy for the locations in the model), smuggler or self and official or unofficial. Combinations of these define the different states and the possible transitions between them are described in Figure 6.6 and represented in Figure 6.7. Figure 6.7 shows the heterogeneity in the model through structural zeros, including the use of smugglers if aiming for Destination A, choosing an unofficial crossing for Destination A if holding a passport, and choosing to cross the border alone if going to Destination

B. Through the presence of nodes with a single emanating edge, it is possible to represent deterministic transitions.

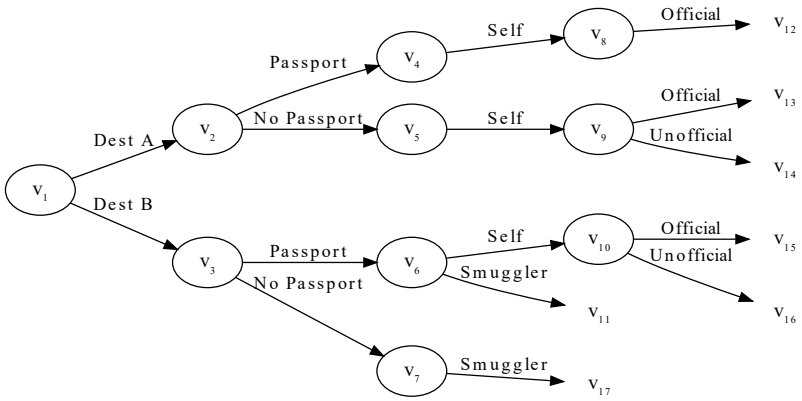


Figure 6.7: Event tree of border crossing example.

The staged tree for this part of the ABM can be seen in Figure 6.8. This is a small part of a larger staged tree representing the entire migration process- from decision to migrate to arrival at their destination. The probabilities of preference of destinations or holding a passport are greyed out as they are not decided in this part of the model: they will have already been allocated in a previous part of the model. However, the ABM in Figure 6.6 identifies these variables as defining the border crossing decision, hence their inclusion in Figure 6.8.

The tree given in Figure 6.8 is not a staged tree of the whole process and is just a sub-tree in the whole process that will appear multiple times, in any sequence of events where a transport decision has not been made previously.

In this sub-tree, no two nodes are in the same stage. However, whenever this sub-tree appears in the larger model, each vertex in white ($v_4 - v_{10}$) will be in the same stage as their corresponding vertices in the other sub-trees.

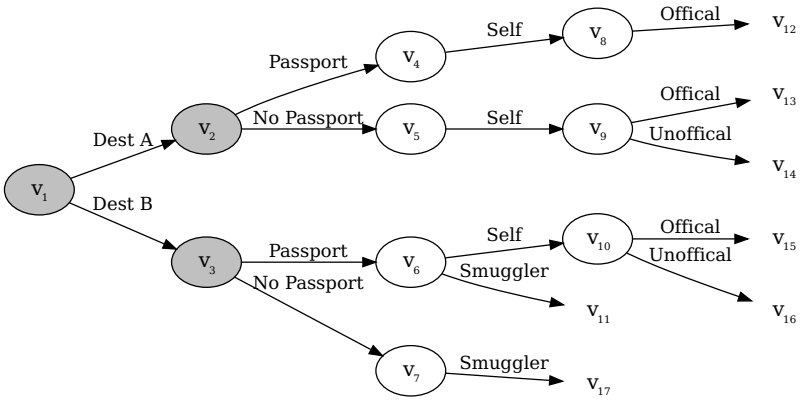


Figure 6.8: Staged tree of border crossing example.

This shows that we are able to use the ABM as a tool to elicit part of a staged tree, providing a graphical representation of the decision process which could be embellished as described in Subsection 6.3.

Out of Scope

Some parts of the MyTh MAP-IN model are not suitable for modelling with a CEG. For example, in the first stage, planning, a key element is the individual wealth fluctuations that occur and how this drives migrant motivation. These wealth fluctuations are not capable of being integrated into a CEG as they are modelled as a Markov process on a continuous variable, mostly providing small perturbations to vary the number of individuals in the model that decide to migrate.

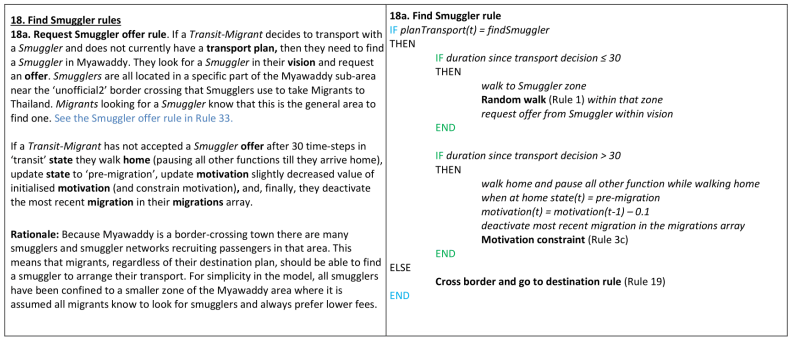


Figure 6.9: Part of the MyTh MAP-IN ABM, describing smuggling decisions. Taken from McAlpine [2021].

Example 53 (Smuggler example) *Figure 6.9 describes the process in the ABM for how an agent finds a smuggler (18a), given that they have decided to. This involves walking to the 'smuggler zone' where the smugglers are based, random walk-*

ing in that area and requesting an offer from a smuggler should they find one. This rule also says that an offer is not received over a certain time period (30 days), the individual stops their migration and returns home.

The core component of this process– finding a smuggler– could easily be modelled using a CEG by directly modelling the probability that someone who is looking for a smuggler finds one.

If time is a key factor in this model, using a CT-DCEG [Shenvi and Smith, 2020b] with holding times associated as to whether a smuggler is found or not could faithfully represent this element of the ABM. A distribution of the length of holding time would be required in this case if we are to fully represent the ABM; this holding time distribution would have all values less than 30 in the instance when a smuggler is found.

6.7 Discussion

In this chapter, we have demonstrated that we are able to use ABMs as a tool to elicit CEGs. The benefits of this transformation are clear: it provides a compact representation of its independence statements, directly from the topology of the graph. This is valuable in identifying whether the model is making a plausible set of assumptions and making the independence structure accessible to be understood by those without a mathematical background, such as policymakers. The transformation into a CEG also allows for a natural conversion into a Bayesian framework with additional benefits: improved uncertainty quantification, Bayesian inference with available data and Bayesian model selection.

This works also highlights the benefits of considering different modelling methodologies the different modelling perspectives can be considered. Comparing between different model classes can be a helpful iterative process in selecting the most effective way to model any process. For example, by considering the CEG, it brings to the forefront what the pathways in the model are and the dependence structure underlying the transitions.

Whilst this chapter specifically focuses on migration, CEGs have many potential applications in other domains where ABMs have been used to represent egocentric processes, such as dietary, voting or criminal behaviour. This research reflects work in progress; further investigation is needed to extend this methodology and increase the scope of ABMs that it applies to. Engaging with these ideas will provide many opportunities for future research to build upon the work presented in this paper, enabling for more full and direct CEG-like representations of a wide

class of ABMs.

Chapter 7

The Posterior Equivalence Principle for Chain Event Graphs

The benefit of informative priors was shown in Chapters 4 and 6 for managing model uncertainty and model selection respectively. In a Bayesian analysis, when using expert judgement in the form of priors, over the model space and on the hyperparameters, it is important that they are used consistently. However, default setting of these prior model probabilities and distributional parameters are usually made without reference to each other.

In this chapter, we define a desirable condition, the Posterior Equivalence Principle (PEP), which guarantees a consistency between priors. We also show how we can set a prior over the set of models to satisfy this condition.

We begin this chapter with a motivational example in Section 7.1, identifying the issues with default prior setting and how this can impact models. Next, in Section 7.2, we define the PEP as a method for consistently setting priors over the edges. We then prove how we can satisfy this principle through the choice of model prior, demonstrating it through an example in Section 7.3. Finally, in Section 7.4, we discuss the benefits of this approach and outline future work. We then apply this principle to the CEG class and return to the initial motivating example to demonstrate its functionality.

7.1 Introduction

Example 54 (Motivating Example) *Suppose we have some data from a school of 600 students about their chosen lunch options. The data contains:*

- *Sex of child (Boy, girl),*
- *Lunch choice (Healthy, unhealthy).*

This data is represented in the event tree in Figure 7.1.

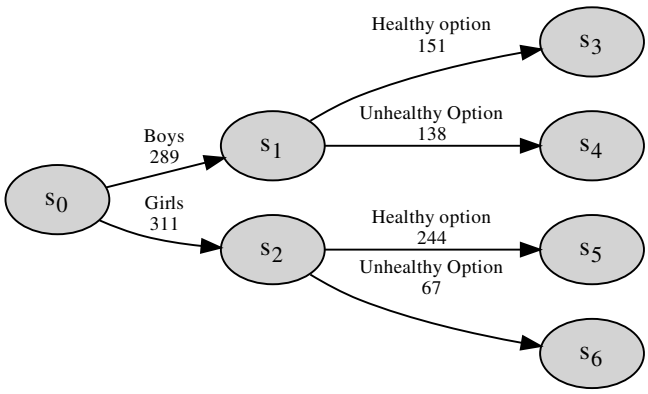


Figure 7.1: Event tree of the school's options data with edge counts

Suppose we have some elicited information about the edge probabilities to use as a prior. These could be elicited by considering experts' judgements as phantom samples observed through the process as recommended by Collazo et al. [2018] and Heckerman et al. [1995], as in Section 3.3.1. Suppose for vertex s_0, s_1 and s_2, these are $Beta(2500, 2500)$, $Beta(1250, 1250)$, $Beta(1250, 1250)$ respectively. These have the same strength (effective sample size) in influencing the posterior edge probabilities as observing 5000 students with an equal number of boys and girls who are also equally likely to choose each lunch option. This is a very strong prior relative to the amount of data observed.

We set the hyperstage such that we only consider situations being in the same stage if they correspond to the same event as detailed in Section 3.4.1. Therefore, there are two possible CEGs for this event tree: either s_1 and s_2 are in the same stage, or they are not. We set an equal prior probability over each CEG in the equivalence class as typically done, for example by Shenvi [2021]. For a fixed ordering of

variables or events, each different staging of a CEG is in its own equivalence class. Therefore, this is to assume a priori that each of the possible models are equally likely (a uniform prior). In this instance, the likelihood of model $\mathcal{C}$ is given by:

$$p(\boldsymbol{y}|\mathcal{C}) = \prod_{i=1}^{K} \left[\frac{\Gamma(\bar{\boldsymbol{\alpha}}_i)}{\Gamma(\bar{\boldsymbol{\alpha}}_i^*)} \prod_{j=1}^{k_i} \frac{\Gamma(\alpha_{ij}^*)}{\Gamma(\alpha_{ij})} \right], \tag{7.1}$$

as in Section 3.3, which can alternatively be written as

$$p(\boldsymbol{y}|\mathcal{C}) = \prod_{i=1}^{K} \left[\frac{\Gamma(\bar{\boldsymbol{\alpha}}_i)}{\prod_{j=1}^{k_i} \Gamma(\alpha_{ij})} \times \frac{\prod_{j=1}^{k_i} \Gamma(\alpha_{ij}^*)}{\Gamma(\bar{\boldsymbol{\alpha}}_i^*)} \right]. \tag{7.2}$$

In this equation, the left part of the product is the contribution from the prior over the edges and the right part is from the combination of the posterior.

In the example, the BF of the situations in the same stage, $\mathcal{C}'$, compared to them in different stages, $\mathcal{C}$, is:

$$\begin{aligned}
p(\boldsymbol{y}|\mathcal{C}) =& \left[\frac{\Gamma(5000)}{\Gamma(5000+600)} \frac{\Gamma(2500+289)\Gamma(2500+311)}{\Gamma(2500)^2} \right] \\
& \left[\frac{\Gamma(2500)}{\Gamma(2500+289)} \frac{\Gamma(1250+151)\Gamma(1250+138)}{\Gamma(1250)^2} \right] \\
& \left[\frac{\Gamma(2500)}{\Gamma(2500+311)} \frac{\Gamma(1250+244)\Gamma(1250+67)}{\Gamma(1250)^2} \right] \\
=& \frac{\Gamma(5000)}{\Gamma(5600)} \frac{\Gamma(1401)\Gamma(1388)\Gamma(1494)\Gamma(1317)}{\Gamma(1250)^4} \\
p(\boldsymbol{y}|\mathcal{C}') =& \left[\frac{\Gamma(5000)}{\Gamma(5000+600)} \frac{\Gamma(2500+289)\Gamma(2500+311)}{\Gamma(2500)^2} \right] \\
& \left[\frac{\Gamma(5000)}{\Gamma(5000+600)} \frac{\Gamma(2500+151+244)\Gamma(2500+138+67)}{\Gamma(2500)^2} \right] \\
=& \frac{\Gamma(5000)^2}{\Gamma(5600)^2} \frac{\Gamma(2789)\Gamma(2811)\Gamma(2895)\Gamma(2705)}{\Gamma(2500)^4} \\
BF(\mathcal{C}', \mathcal{C}) =& \frac{p(\boldsymbol{y}|\mathcal{C}')}{p(\boldsymbol{y}|\mathcal{C})} \\
=& \frac{\Gamma(5000)}{\Gamma(5600)} \frac{\Gamma(1250)^4}{\Gamma(2500)^4} \frac{\Gamma(2789)\Gamma(2811)\Gamma(2895)\Gamma(2705)}{\Gamma(1401)\Gamma(1388)\Gamma(1494)\Gamma(1317)}.
\end{aligned}$$

This gives a BF of 0.0975. Therefore, the model with these situations apart would be the MAP model and, even when performing BMA, the model with the

situations together would have a smaller relative weighting.

However, if we have strong prior information reflected on the edges which represents that s_1 and s_2 having the same distribution, does it not make more sense that they should be in the same stage?

The solution to the problem raised in this example is clearly that the assumption that all of the models should be set as a default to equally likely *a priori* is very questionable when there is strong prior information about the transition probabilities. This raises the question of how the prior over the set of models could be set in a way that is consistent with the prior conditional transition probability parameters.

7.2 Posterior Equivalence Principle (PEP)

Little work has been done on setting the priors over the set of models for CEGs. That which has been done, such as in Collazo and Smith [2016], is not focused on prior setting for expert elicitation but for the soundness of model selection, using non-local priors. Here, we wish to set priors over the model space that are consistent with the priors set over the edges.

More formally, we define this consistency as follows:

Definition 55 (Posterior Equivalence Principle (PEP)) *Given any two CEGs, $\mathcal{C}$ and $\mathcal{C}'$, that represent the same staging on the same event tree in the same set of models $\mathcal{S}$ – but with different priors, $\boldsymbol{\alpha}$, and data, $\boldsymbol{y}$,– that share constant α^*_{ij} for all edges ij, as the sum of the prior, α_{ij}, and the data, y_{ij}, over that edge:*

$$\alpha^*_{ij} = \alpha_{ij} + y_{ij}, \tag{7.3}$$

we say that the PEP is satisfied if the joint distribution of the model and the data are the same, $p(\mathcal{C}, y) = p(\mathcal{C}', y)$.

As shown in Section 3.3, for a CEG, the posterior model is made of a combination of florets each with distribution $\boldsymbol{\theta_i}$:

$$p(\boldsymbol{\theta}_i|\boldsymbol{y}_i, \mathcal{C}) = \prod_{i=1}^{K} \frac{\Gamma(\bar{\boldsymbol{\alpha}}^*_{ij})}{\prod_{j=1}^{k_j} \Gamma(\alpha^*_{ij})} \prod_{j=1}^{k_i} \theta_{ij}^{\alpha^*_{ij}-1}. \tag{7.4}$$

Apart from the stage and tree structure which determine K and k_i, the set of posterior conditional transition probability parameters is determined uniquely by the vectors $\boldsymbol{\alpha}^*_{ij}$.

Therefore, PEP is the condition that the joint distribution of the model and the data, the general BD score, is uniquely determined by the posterior conditional transition probability parameters.

7.3 Satisfying the PEP

To satisfy PEP for CEGs, we propose the following prior for $M_i \in \mathcal{S}$:

$$p(M_i) \propto \prod_{i=1}^{K} \left[\frac{\Gamma(\bar{\boldsymbol{\beta}}_i)}{\Gamma(\bar{\boldsymbol{\alpha}}_i)} \prod_{j=1}^{k_i} \frac{\Gamma(\alpha_{ij})}{\Gamma(\beta_{ij})} \right]. \tag{7.5}$$

We define this prior as the PEP prior. Here, as before, α_{ij} is the prior parameters for the transition probabilities and β_{ij} is the uniform model edge prior, a prior edge count that is believed to correspond to all models being equally likely; this could be set to correspond to default weakly informative priors, such as those suggested in Neapolitan [2003]. We must condition that for all i and j, $\beta_{ij} \leq \alpha_{ij}$. When $\beta_{ij} = \alpha_{ij}$ for all i and j, this gives the uniform prior.

We will now demonstrate why this setting of the prior over the space of CEGs satisfies the PEP. The PEP-prior in Equation (7.5) gives the following general BD score for model M_i:

$$\begin{aligned} p(M_i, \boldsymbol{y}) &= p(M_i) p(\boldsymbol{y}|M_i) \\ &\propto \prod_{i=1}^{K} \left[\frac{\Gamma(\bar{\boldsymbol{\beta}}_i)}{\Gamma(\bar{\boldsymbol{\alpha}}_i)} \prod_{j=1}^{k_i} \frac{\Gamma(\alpha_{ij})}{\Gamma(\beta_{ij})} \right] \prod_{i=1}^{K} \left[\frac{\Gamma(\bar{\boldsymbol{\alpha}}_i)}{\Gamma(\bar{\boldsymbol{\alpha}}_i^*)} \prod_{j=1}^{k_i} \frac{\Gamma(\alpha_{ij}^*)}{\Gamma(\alpha_{ij})} \right] \\ &= \prod_{i=1}^{K} \left[\frac{\Gamma(\bar{\boldsymbol{\beta}}_i)}{\Gamma(\bar{\boldsymbol{\alpha}}_i^*)} \prod_{j=1}^{k_i} \frac{\Gamma(\alpha_{ij}^*)}{\Gamma(\beta_{ij})} \right]. \end{aligned} \tag{7.6}$$

This score depends only on the choice of $\boldsymbol{\beta}_i$ for the PEP-prior and the posterior conditional transition probability parameters. Therefore, if we use the same $\boldsymbol{\beta}_i$ values for the PEP-priors, the PEP is satisfied.

Example 56 (Motivating example continued) *We return to our example of lunch choices of school children using the same data and parameter prior as before.*

We use the proposed structural prior from the previous section, the PEP-prior, with uniform model edge priors $\boldsymbol{\beta}_i$, being that specified in Neapolitan [2003]: total prior weight is equal to the maximum number of outgoing edges. These uniform

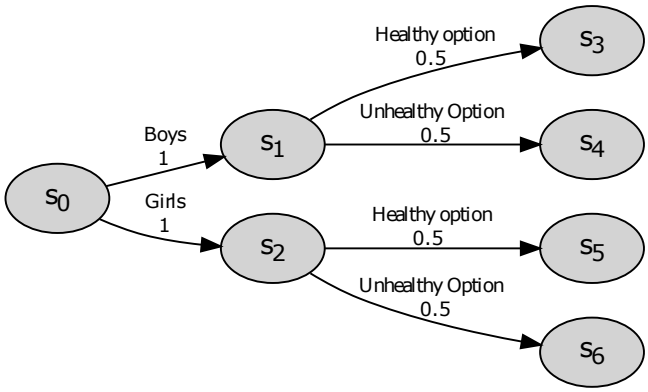


Figure 7.2: Event tree of the school's options with beta values along the edge that are judged to be equivalent to a uniform prior over the set of models.

model edge priors are shown in Figure 7.2.

This gives a PEP-prior over the possible models of:

$$p(C) \propto \left[\frac{\Gamma(2)}{\Gamma(5000)}\frac{\Gamma(2500)^2}{\Gamma(1)^2}\right]\left[\frac{\Gamma(1)}{\Gamma(2500)}\frac{\Gamma(1250)^2}{\Gamma(0.5)^2}\right]^2, \tag{7.7}$$

$$p(C') \propto \left[\frac{\Gamma(2)}{\Gamma(5000)}\frac{\Gamma(2500)^2}{\Gamma(1)^2}\right]\left[\frac{\Gamma(2)}{\Gamma(5000)}\frac{\Gamma(2500)^2}{\Gamma(1)^2}\right]. \tag{7.8}$$

This gives a prior probability of 0.986 of the situations being combined[1].

[1]Calculated by taking their log and then normalising.

$$
\begin{aligned}
p(y|\mathcal{C}) =& \left[\frac{\Gamma(2)}{\Gamma(5600)}\frac{\Gamma(2500+289)\Gamma(2500+311)}{\Gamma(1)^2}\right] \\
& \left[\frac{\Gamma(1)}{\Gamma(2500+289)}\frac{\Gamma(1250+151)\Gamma(1250+138)}{\Gamma(0.5)^2}\right] \\
& \left[\frac{\Gamma(1)}{\Gamma(2500+311)}\frac{\Gamma(1250+244)\Gamma(1250+67)}{\Gamma(0.5)^2}\right] \\
=& \frac{\Gamma(2)}{\Gamma(5600)}\frac{\Gamma(1401)\Gamma(1388)\Gamma(1494)\Gamma(1317)}{\Gamma(0.5)^4} \\
p(y|\mathcal{C}') =& \left[\frac{\Gamma(2)}{\Gamma(5000+600)}\frac{\Gamma(2500+289)\Gamma(2500+311)}{\Gamma(1)^2}\right] \\
& \left[\frac{\Gamma(2)}{\Gamma(5000+600)}\frac{\Gamma(2500+151+244)\Gamma(2500+138+67)}{\Gamma(1)^2}\right] \\
=& \frac{\Gamma(2)^2}{\Gamma(5600)^2}\frac{\Gamma(2789)\Gamma(2811)\Gamma(2895)\Gamma(2705)}{\Gamma(1)^4} \\
BF(\mathcal{C}', \mathcal{C}) =& \frac{p(y|\mathcal{C}')}{p(y|\mathcal{C})} \\
=& \frac{\Gamma(2)}{\Gamma(5600)}\frac{\Gamma(0.5)^4}{\Gamma(1)^4}\frac{\Gamma(2789)\Gamma(2811)\Gamma(2895)\Gamma(2705)}{\Gamma(1401)\Gamma(1388)\Gamma(1494)\Gamma(1317)}
\end{aligned}
$$

This gives a BF of 6.79, meaning the situations should be combined.

7.4 Discussion

Here, we have defined an invariant condition, PEP, that can be used in model selection of CEGs when expert elicitation is involved. This condition means that the priors over the model space do not need to be elicited separately from the effective sample size of the florets.

The setting of the PEP-prior raises an interesting question of what $\boldsymbol{\beta}_i$ values should be chosen. However, we would argue that this is not a new issue and simply formalises the idea of what should be used as a default weakly informative prior that corresponds to a uniform prior over the set of models, as used when there is no expert elicitation.

Outside of expert elicitation, this work has application to dynamic CEGs as described in Freeman and Smith [2011b], in which the modelled process is fixed but the staging can change over time. Through the use of a discount factor, PEP allows for a way of exploring the change of staging over different time periods whilst

maintaining consistency over how previous years' data was treated.

Chapter 8

Discussion

In Section 8.1, we summarise the main contributions of this thesis. Section 8.2 gives details on work which continues beyond the completion of this thesis, in terms of software development related to the `cegpy` package and short-term avenues for future research.

In additional to those outlined in Section 8.2, each research chapter concluded with a discussion regarding the potential further avenues for research related to that particular field.

8.1 Summary of the contributions of the Thesis

The four research chapters of this thesis include significant contributions to the CEG research community. In addition to the individual elements of research presented, we provided code for the new `cegpy` package, increasing the accessibility of non-stratified CEGs to applied statisticians and providing opportunities to further expand understanding of CEGs. The `cegpy` package was used as a tool to develop the research in Chapters 4 and 5.

In Chapter 4, we presented a framework for performing BMA for CEGs using Occam's window and demonstrated the benefits of this approach compared to using a MAP estimate. We proved that, by sampling the model space, we can obtain a consistent estimator of the BMA and defined one such sampler. In our analysis of BMA's benefits, we defined the most refined union and coarsest intersection of a set of CEGs and demonstrated how this can be used to interpret the output of a BMA.

In Chapter 5, we defined the totally-ordered hyperset and hyperstage. We presented a model selection algorithm that scales quadratically with the number of situations, MPC, using the totally-ordered hyperstage. We defined the BCEG, an

expansion of the model space of CEGs capable of representing more complicated relationships between events. Through the use of a number of datasets, we demonstrated that MPC outputted comparably scoring models to AHC in much faster time and that BCEGs are able to obtain better scoring models than those in a typical CEG model space. Through a worked example, we demonstrated the increased explainability using a BCEG.

In Chapter 6, we demonstrated the differences in representation of a flow diagram and a CEG. We also detailed the benefits of eliciting a CEG from an ABM and illustrated how, for an existing ABM, parts of it can easily and effectively translated into a CEG. We also highlighted future areas of research in which a larger class of ABMs, with a range of features, can be used to elicit CEG-based models. We finished this chapter by demonstrating this approach in the domain of migration research.

In Chapter 7, we defined an invariance condition, the PEP, to ensure that priors over the set of models and priors of the parameters of the models are set consistently. We also demonstrate how the PEP can be satisfied through the choice of model prior, the PEP-prior, that depends on a choice of prior edge counts that corresponds to a uniform prior being set over the model space.

8.2 Future work

Below we detail some other potential avenues for future research:

- As discussed in Section 3.6, the `cegpy` package was developed to aid the use of CEGs to a wider range of users. In order for this package to provide a useful tool for the community, it is important that it both incorporates a range of describable features, including methodology advances, and is regularly maintained, as demonstrated by the popularity of BNs supported by a range of well maintained and easy to use software. This thesis includes methodological advancements that were demonstrated by building on the `cegpy` package. Adding this functionality to `cegpy` and other existing CEG methodology is an important next step in furthering the use of CEGs.

- The research in this thesis focuses on methodological advancements for a fixed tree. An obvious extension is to consider how to perform BMA when there is not a fixed ordering over the sequence of events. This would involve exploring the statistical equivalence class of CEGs and determining cardinality in order to set the model's priors in a way such that models that are represented

multiple times in the model space are not *a priori* more likely.

- As discussed in Section 6.3.1, there is currently no algorithm that traverses the equivalence class of CEGs in order to output the simplest representation of a CEG. The existence of such an algorithm could provide compact representations of elicited situations to an extent further than the standard CEG by removing florets with a single edge and combining florets when no staging information was lost.

- Further extensions on embellishing ABMs of migration is planned with domain experts and current collaborators. This work will involve modelling the relationship between migration and sexual exploitation. Although this thesis has focused on the technical challenges encountered over the course of this work, there are many exciting avenues of work springing from these.

- During my PhD, I also engaged in another project where, in collaboration with Jim Smith, we used a CEG in a dynamic hierarchical model, linking these to another class of models called flow graphs. This contained some novel work, linking CEGs and flow graphs to decision analysis. Over the coming months, we plan to extend these model classes so that they might apply to a wider domain base.

- Recent research has seen many developments in the methodology surrounding CEGs, demonstrating them as being widely applicable due to the asymmetric relations they can represent. A key focus of future research should be to work with an increased range of domains, engaging with experts. This will allow the benefits of CEGs to be utilised in a wider domain area and facilitate the opportunity to encounter further methodological challenges that arise from the domains being modelled, thereby improving the CEG class as a whole.

Thank you for reading this thesis.

Bibliography

Hirotugu Akaike. A New Look at the Statistical Model Identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, 1974.

William Alonso. *Systemic and Log-Linear Models: From Here to There, Then to Now, and This to That*. Harvard University, Center for Population Studies, 1986.

Li An, Volker Grimm, Abigail Sullivan, B.L. Turner II, Nicolas Malleson, Alison Heppenstall, Christian Vincenot, Derek Robinson, Xinyue Ye, Jianguo Liu, Emilie Lindkvist, and Wenwu Tang. Challenges, Tasks, and Opportunities in Modeling Agent-Based Complex Systems. *Ecological Modelling*, 2021. URL https://www.sciencedirect.com/science/article/pii/S030438002100243X.

Stig K. Andersen, Kristian G. Olesen, Finn Verner Jensen, and Frank Jensen. HUGIN-A Shell for Building Bayesian Belief Universes for Expert Systems. In *IJCAI*, volume 89, pages 1080–1085, 1989.

Paul E. Anderson and Jim Q. Smith. A Graphical Framework for Representing the Semantics of Asymmetric Models. Technical report, Citeseer, 2005.

Jennifer Badham. Agent-Based Modelling for the Self Learner Tutorials Edition, May 2020. URL https://research.criticalconnections.com.au/ABMBook/.

Lorna M. Barclay, Jane L. Hutton, and Jim Q. Smith. Refining a Bayesian Network using a Chain Event Graph. *International Journal of Approximate Reasoning*, 54 (9):1300–1309, 2013.

Lorna M. Barclay, Jane L. Hutton, and Jim Q. Smith. Chain Event Graphs for Informed Missingness. *Bayesian Analysis*, 9(1), 2014. doi: 10.1214/13-ba843.

Lorna M. Barclay, Rodrigo A. Collazo, Jim Q. Smith, Peter A. Thwaites, and Ann E. Nicholson. The Dynamic Chain Event Graph. *Electronic Journal of Statistics*, 9 (2):2130–2169, 2015.

Martine J. Barons, Sophia K. Wright, and Jim Q. Smith. Eliciting Probabilistic Judgements for Integrating Decision Support Systems. In *Elicitation*, pages 445–478. Springer, 2018.

BayesFusion, LLC. *GeNIe Modeler*, 2022. Version 4.0.

José M. Bernardo and Adrian F. M. Smith. *Bayesian Theory*, volume 405. John Wiley & Sons, 2009.

Jakub Bijak. *Towards Bayesian Model-Based Demography: Agency, Complexity and Uncertainty in Migration Studies*. Springer Nature, 2022.

Susanne Gammelgaard Bottcher and Claus Dethlefsen. *deal: Learning Bayesian Networks with Mixed Variables*, 2018. R pkg Version 1.2-39.

F. Oliver Bunnin and Jim Q. Smith. A Bayesian Hierarchical Model for Criminal Investigations. *Bayesian Analysis*, 16(1):1–30, 2021.

F. Oliver Bunnin, Aditi Shenvi, and Jim Q. Smith. Network Modelling of Criminal Collaborations with Dynamic Bayesian Steady Evolutions. arXiv preprint arXiv:2007.04410, 2020.

Rafael Cano, Carmen Sordo, and José M. Gutiérrez. Applications of Bayesian Networks in Meteorology. *Advances in Bayesian networks*, pages 309–328, 2004.

Federico Carli, Manuele Leonelli, Eva Riccomagno, and Gherardo Varando. The R package stagedtrees for Structural Learning of Stratified Staged Trees, 2020. arXivpreprint. arXiv2004.06459.

Rodrigo Collazo and Pier Taranti. *ceg: Chain Event Graph*, 2017. R pkg Version 0.1.0.

Rodrigo A. Collazo and Jim Q. Smith. A New Family of Non-Local Priors for Chain Event Graph Model Selection. *Bayesian Analysis*, 11(4):1165 – 1201, 2016. doi: 10.1214/15-BA981. URL https://doi.org/10.1214/15-BA981.

Rodrigo A. Collazo, Christiane Görgen, and Jim Q. Smith. *Chain Event Graphs*. CRC Press, 2018.

Rodrigo Abrunhosa Collazo. *The Dynamic Chain Event Graph*. PhD thesis, The University of Warwick, 2017.

National Research Council, Committee on Population, et al. *Beyond Six Billion: Forecasting the World's Population*. National Academies Press, 2000.

Robert G. Cowell and Jim Q. Smith. Causal Discovery through MAP Selection of Stratified Chain Event Graphs. *Electronic Journal of Statistics*, 8(1):965–997, 2014.

James Cussens. GOBNILP: Learning Bayesian Network Structure with Integer Programming. In *Intern. Conf. on Probab. Graph. Models*, pages 605–608. PMLR, 2020.

Jeremy Dale, Veronica Nanton, Theresa Day, Patricia Apenteng, Celia Janine Bernstein, Gillian Grason Smith, Peter Strong, and Rob Procter. Care Companion: A Mixed Methods, Real World Evaluation of the use of an Online Resource to Support Informal Carers. JMIR preprint, 2022.

Brett Drury, Jorge Valverde-Rebaza, Maria-Fernanda Moura, and Alneu de Andrade Lopes. A Survey of the Applications of Bayesian Networks in Agriculture. *Engineering Applications of Artificial Intelligence*, 65:29–42, 2017.

Eliana Duarte and Liam Solus. Representation of Context-Specific Causal Mdels with Observational and Interventional Data. arXiv preprint. arXiv:2101.09271, 2021.

Bruce Edmonds. Different Modelling Purposes. *Simulating Social Complexity: A Handbook*, pages 39–58, 2017.

Frank Eibe, Mark A. Hall, and Ian H. Witten. *The WEKA Workbench.* Morgan Kaufmann Publishers, 2016.

Barbara Entwisle, Nathalie E. Williams, Ashton M. Verdery, Ronald R. Rindfuss, Stephen J. Walsh, George P. Malanson, Peter J. Mucha, Brian G. Frizzelle, Philip M. McDaniel, Xiaozheng Yao, et al. Climate Shocks and Migration: An Agent-Based Modeling Approach. *Population and environment*, 38(1):47–71, 2016.

Joshua M. Epstein and Robert Axtell. *Growing Artificial Societies: Social Science from the Bottom Up.* Brookings Institution Press, 1996.

Robert Epstein. The Principle of Parsimony and Some Applications in Psychology. *Journal of Mind and Behavior*, 5, 1984.

Marta Bivand Erdal and Ceri Oeppen. Forced to Leave? The Discursive and Analytical Significance of Describing Migration as Forced and Voluntary. *Journal of Ethnic and Migration Studies*, 44(6):981–998, 2018.

J. Doyne Farmer and Duncan Foley. The Economy Needs Agent-Based Modelling. *Nature*, 460(7256):685–686, 2009.

David M. Fergusson, L. John Horwood, and Frederick T. Shannon. Social and Family Factors in Childhood Hospital Admission. *Journal of Epidemiology & Community Health*, 40(1):50–58, 1986.

Tiago M. Fragoso, Wesley Bertoli, and Francisco Louzada. Bayesian Model Averaging: A Systematic Review and Conceptual Classification. *International Statistical Review*, 86(1):1–28, 2018.

Guy Freeman and Jim Q. Smith. Bayesian MAP Model Selection of Chain Event Graphs. *Journal of Multivariate Analysis*, 102(7):1152–1165, 2011a.

Guy Freeman and Jim Q. Smith. Dynamic Staged Trees for Discrete Multivariate Time Series: Forecasting, Model Selection and Causal Analysis. *Bayesian Anal.*, 6(2), 2011b. doi: 10.1214/11-ba610.

Zhaohao Fu and Lingxin Hao. Agent-Based Modeling of China's Rural–Urban Migration and Social Network Structure. *Physica A: Statistical Mechanics and its Applications*, 490:1061–1075, 2018.

José Manuel Galán, Luis R. Izquierdo, Segismundo S. Izquierdo, José Ignacio Santos, Ricardo del Olmo, Adolfo López-Paredes, and Bruce Edmonds. Errors and Artefacts in Agent-Based Modelling. *Journal of Artificial Societies and Social Simulation*, 12(1):1, 2009. ISSN 1460-7425. URL https://www.jasss.org/12/1/1.html.

Nigel Gilbert. *Agent-Based Models*, volume 153. Sage Publications, 2019.

Nigel Gilbert and Pietro Terna. How to Build and use Agent-Based Models in Social Science. *Mind & Society*, 1:57–72, 2000.

Christiane Görgen, Aida Maraj, and Lisa Nicklasson. Staged Tree Models with Toric Structure. *Journal of Symbolic Computation*, 113:242–268, 2022.

Dennis M. Gorman, Jadranka Mezic, Igor Mezic, and Paul J. Gruenewald. Agent-Based Modeling of Drinking Behavior: A Preliminary Model and Potential Applications to Theory and Practice. *American Journal of Public Health*, 96(11): 2055–2060, 2006.

Michael J. Greenwood and Gary L. Hunt. The Early History of Migration Research. *International Regional Science Review*, 26(1):3–37, 2003.

Volker Grimm, Eloy Revilla, Uta Berger, Florian Jeltsch, Wolf M. Mooij, Steven F. Railsback, Hans-Hermann Thulke, Jacob Weiner, Thorsten Wiegand, Donald L. DeAngelis, and et al. Pattern-Oriented Modeling of Agent-Based Complex Systems: Lessons from Ecology. *Science*, 2005. URL https://www.science.org/doi/10.1126/science.1116681.

Volker Grimm, Uta Berger, Finn Bastiansen, Sigrunn Eliassen, Vincent Ginot, Jarl Giske, John Goss-Custard, Tamara Grand, Simone K. Heinz, Geir Huse, and et al. A Standard Protocol for Describing Individual-Based and Agent-Based Models. *Ecol. Modell.*, 198(1-2):115–126, 2006. doi: {10.1016/j.ecolmodel.2006.04.023}.

Elizabeth R. Groff, Shane D. Johnson, and Amy Thornton. State of the Art in Agent-Based Modeling of Urban Crime: An Overview. *Journal of Quantitative Criminology*, 35(1):155–193, 2019.

Christiane Görgen and Jim Q. Smith. Equivalence Classes of Staged Trees. *Bernoulli*, 24(4A):2676 – 2692, 2018. doi: 10.3150/17-BEJ940.

Scott Heckbert, Tim Baynes, and Andrew Reeson. Agent-Based Modeling in Ecological Economics. *Ann. N. Y. Acad. Sci.*, 2010. URL https://nyaspubs.onlinelibrary.wiley.com/doi/10.1111/j.1749-6632.2009.05286.x.

David Heckerman, Dan Geiger, and David M. Chickering. Learning Bayesian Networks: The Combination of Knowledge and Statistical Data. *Machine Learning*, 20(3):197–243, 1995. doi: 10.1007/bf00994016.

Franziska Hinkelmann, David Murrugarra, Abdul Salam Jarrah, and Reinhard Laubenbacher. A Mathematical Framework for Agent Based Models of Complex Biological Networks. *Bull. Math. Biol.*, 73(7):1583–1602, 2010. doi: {10.1007/s11538-010-9582-8}.

Max Hinne, Quentin F. Gronau, Don van den Bergh, and Eric-Jan Wagenmakers. A Conceptual Introduction to Bayesian Model Averaging. *Advances in Methods and Practices in Psychological Science*, 3(2):200–215, 2020.

Ronald Arthur Howard and James E. Matheson. *Readings on the Principles and Applications of Decision Analysis: Professional Collection*, volume 2. Strategic Decisions Group, 1981.

Linwood D. Hudson, Bryan S. Ware, Kathryn B. Laskey, and Suzanne M. Mahoney. An Application of Bayesian Networks to Antiterrorism Risk Management for Military Planners. *George Mason University*, 2005.

Conor Hughes, Peter Strong, and Aditi Shenvi. On Bayesian Dirichlet Scores for Staged Trees and Chain Event Graphs, 2022. arXiv.2206.15322.

HuginExpert. *HUGIN*, 2022. Version 9.2.

David Rios Insua and Simon French. *E-Democracy: a Group Decision and Negotiation Perspective*. Springer Science Business Media B.V., 2010.

International Labour Organisation. Global Estimates of Modern Slavery: Forced Labour and Forced Marriage. Technical report, International Labour Organisation, 2017.

Manfred Jaeger. Probabilistic Decision Graphs—Combining Verification and AI Techniques for Probabilistic Inference. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 12(supp01):19–42, 2004.

Finn V. Jensen and Thomas Dyhre Nielsen. *Bayesian Networks and Decision Graphs*, volume 2. Springer, 2007.

David Kaplan and Chansoon Lee. Optimizing Prediction using Bayesian Model Averaging: Examples Using Large-Scale Educational Assessments. *Evaluation Review*, 42(4):423–457, 2018. doi: 10.1177/0193841X18761421. URL https://doi.org/10.1177/0193841X18761421. PMID: 29642717.

Robert E. Kass and Adrian E. Raftery. Bayes Factors. *Journal of the American Statistical Association*, 90(430):773–795, 1995.

Albert Kraler and David Reichel. Measuring Irregular Migration and Population Flows– What Available Data can Tell. *International Migration*, 49(5):97–128, 2011.

Steffen L. Lauritzen. *Graphical Models*. Clarendon Press, 1996.

Manuele Leonelli and Gherardo Varando. Highly Efficient Structural Learning of Sparse Staged Trees. In *Proceedings of The 11th International Conference on Probabilistic Graphical Models*, volume 186 of *Proceedings of Machine Learning Research*, pages 193–204. PMLR, 05–07 Oct 2022. URL https://proceedings.mlr.press/v186/leonelli22a.html.

Hannah Lewis, Peter Dwyer, Stuart Hodkinson, and Louise Waite. Hyper-Precarious Lives: Migrants, Work and Forced Labour in the Global North. *Progress in Human Geography*, 39(5):580–600, 2015. doi: 10.1177/0309132514548303.

Douglas A. Luke and Katherine A. Stamatakis. Systems Science Methods in Public Health: Dynamics, Networks, and Agents. *Annual review of public health*, 33:357, 2012.

Jaakko Luttinen. BayesPy: Variational Bayesian Inference in Python. *J. of Mach. Learn. Res.*, 17(1):1419–1424, 2016.

David Madigan and Adrian E. Raftery. Model Selection and Accounting for Model Uncertainty in Graphical Models Using Occam's Window. *Journal of the American Statistical Association*, 89(428):1535–1546, 1994.

David Madigan, Jeremy York, and Denis Allard. Bayesian Graphical Models for Discrete Data. *International Statistical Review / Revue Internationale de Statistique*, 63(2):215–232, 1995. ISSN 03067734, 17515823.

David Madigan, Steen A. Andersson, Michael D. Perlman, and Chris T. Volinsky. Bayesian Model Averaging and Model Selection for Markov Equivalence Classes of Acyclic Digraphs. *Communications in Statistics–Theory and Methods*, 25(11): 2493–2519, 1996.

Alys McAlpine. *Mediated Labour Migration in the Myanmar-Thailand Corridor and Precarious Outcomes: a Mixed Methods Social Network Analysis and Agent-Based Model*. PhD thesis, London School of Hygiene & Tropical Medicine, 2021.

Alys McAlpine, Ligia Kiss, Cathy Zimmerman, and Zaid Chalabi. Agent-Based Modeling for Migration and Modern Slavery Research: A Systematic Review. *Journal of Computational Social Science*, 4(1):243–332, 2020. doi: {10.1007/s42001-020-00076-7}.

Daniel McFadden. Econometric Models of Probabilistic Choice. *Structural Analysis of Discrete Data with Econometric Applications*, 1981.

Richard Neapolitan. *Learning Bayesian Networks*. 01 2003. ISBN 9780123704771. doi: 10.1145/1327942.1327961.

Frank Nielsen. *Introduction to HPC with MPI for Data Science*. Springer, 2016.

Norsys Software Corp. *Netica*, 2020. Version 6.08.

Erik P. Nyberg, Ann E. Nicholson, Kevin B. Korb, Michael Wybrow, Ingrid Zukerman, Steven Mascaro, Shreshth Thakur, Abraham Oshni Alvandi, Jeff Riley, Ross Pearson, et al. BARD: A Structured Technique for Group Elicitation of Bayesian Networks to Support Analytic Reasoning. *Risk Anal.*, 42(6):1155–1178, 2022.

David O'Sullivan, James Millington, George Perry, and John Wainwright. Agent-Based Models–because They're Worth It? In *Agent-based models of geographical systems*, pages 109–123. Springer, 2012.

Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

Judea Pearl. *Causality*. Cambridge University Press, 2 edition, 2009. doi: 10.1017/CBO9780511803161.

Howard Raiffa. *Decision Analysis: Introductory Lectures on Choices Under Uncertainty*. Addison-Wesley, 1968.

Ernst Georg Ravenstein. *The Laws of Migration*. 1885.

Sylvia Richardson and Peter J. Green. On Bayesian Analysis of Mixtures with an Unknown Number of Components (with discussion). *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 59(4):731–792, 1997.

Anna Rigosi, Paul Hanson, David P. Hamilton, Matthew Hipsey, James A. Rusak, Julie Bois, Karin Sparber, Ingrid Chorus, Andrew J. Watkinson, Boqiang Qin, et al. Determining the Probability of Cyanobacterial Blooms: The Application of Bayesian Networks in Multiple Lake Systems. *Ecological Applications*, 25(1): 186–199, 2015.

Jule Schulze, Birgit Müller, Jürgen Groeneveld, and Volker Grimm. Agent-Based Modelling of Social-Ecological Systems: Achievements, Challenges, and a Way Forward. *Journal of Artificial Societies and Social Simulation*, 20(2), 2017.

Gideon Schwarz. Estimating the Dimension of a Model. *The Annals of Statistics*, 6 (2):461–464, 1978.

Marco Scutari. Learning Bayesian Networks with the `bnlearn` R package. *Journal of Statistical Software*, 35(3):1–22, 2010.

Marco Scutari. Dirichlet Bayesian Network Scores and the Maximum Relative Entropy Principle. *Behaviormetrika*, 45(2):337–362, 2018.

Ross D. Shachter. Probabilistic Inference and Influence Diagrams. *Operations Research*, 36(4):589–604, 1988.

Glenn Shafer. *The Art of Causal Conjecture*. MIT Press, 1996.

Aditi Shenvi. *Non-Stratified Chain Event Graphs: Dynamic Variants, Inference and Applications*. PhD thesis, 2021.

Aditi Shenvi and Silvia Liverani. Beyond Conjugacy for Chain Event Graph Model Selection. *arXiv preprint arXiv:2211.03427*, 2022.

Aditi Shenvi and Jim Q. Smith. A Bayesian Dynamic Graphical Model for Recurrent Events in Public Health. arXiv preprint arXiv:1811.08872, 2019.

Aditi Shenvi and Jim Q. Smith. Constructing a Chain Event Graph from a Staged Tree. In *Proceedings of 10th European Workshop on Probabilistic Graphical Models*, 2020a.

Aditi Shenvi and Jim Q. Smith. Propagation for Dynamic Continuous Time Chain Event Graphs. arXiv preprint arXiv:2006.15865, 2020b.

Aditi Shenvi, Jim Q. Smith, Robert Walton, and Sandra Eldridge. Modelling with Non-Stratified Chain Event Graphs. In *International Conference on Bayesian Statistics in Action*, pages 155–163. Springer, 2018.

Aditi Shenvi, Francis Oliver Bunnin, and Jim Q. Smith. A Bayesian Decision Support System for Counteracting Activities of Terrorist Groups. *Journal of the Royal Statistical Society Series A: Statistics in Society*, 01 2023. URL https://doi.org/10.1093/jrsssa/qnac019.

Tomi Silander and Tze-Yun Leong. A Dynamic Programming Algorithm for Learning Chain Event Graphs. In *International Conference on Discovery Science*, pages 201–216. Springer, 2013.

Tomi Silander, Teemu Roos, and Petri Myllymäki. Learning Locally Minimax Optimal Bayesian Networks. *International Journal of Approximate Reasoning*, 51(5): 544–557, 2010.

Jim Q. Smith and Paul E. Anderson. Conditional Independence and Chain Event Graphs. *Artificial Intelligence*, 172(1):42–68, 2008.

David J. Spiegelhalter and Steffen L. Lauritzen. Sequential Updating of Conditional Probabilities on Directed Graphical Structures. *Networks*, 20(5):579–605, 1990.

John Q. Stewart. Empirical Mathematical Rules Concerning the Distribution and Equilibrium of Population. *Geographical Review*, 37(3):461–485, 1947.

Peter Strong and Jim Q. Smith. Scalable Model Selection for Staged Trees: Mean-Posterior Clustering and Binary Trees. *arXiv preprint arXiv:2211.07228*, 2022a.

Peter Strong and Jim Q. Smith. Bayesian Model Averaging of Chain Event Graphs for Robust Explanatory Modelling. In *Proceedings of The 11th International Conference on Probabilistic Graphical Models*, volume 186 of *Proceedings of Machine Learning Research*, pages 61–72, 2022b. URL https://proceedings.mlr.press/v186/strong22a.html.

Peter Strong, Aditi Shenvi, Xuewen Yu, K. Nadia Papamichail, Henry P. Wynn, and Jim Q. Smith. Building a Bayesian Decision Support System for Evaluating COVID-19 Countermeasure Strategies. *Journal of the Operational Research Society*, pages 1–13, 2021.

Peter Strong, Alys McAlpine, and Jim Q. Smith. Towards a Bayesian Analysis of Migration Pathways Using Chain Event Graphs of Agent Based Models. In *New Frontiers in Bayesian Statistics: BAYSM 2021*, pages 23–33. Springer, 2022.

Peter Thwaites. Causal Identifiability via Chain Event Graphs. *Artificial Intelligence*, 195:291–315, 2013.

Peter Thwaites, Jim Q. Smith, and Eva Riccomagno. Causal Analysis with Chain Event Graphs. *Artificial Intelligence*, 174(12-13):889–909, 2010.

Peter A Thwaites, Jim Q. Smith, and Robert G. Cowell. Propagation using Chain Event Graphs. In *Proceedings of the Twenty-Fourth Conference on Uncertainty in Artificial Intelligence*, pages 546–553, 2008.

Jin Tian, Ru He, and Lavanya Ram. Bayesian Model Averaging Using the k-best Bayesian Network Structures. In *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence*, UAI'10, page 589–597. AUAI Press, 2010.

Melissa Tracy, Magdalena Cerdá, and Katherine M. Keyes. Agent-Based Modeling in Public Health: Current Applications and Future Directions. *Annual Review of Public Health*, 39, 2018.

Jennifer S. Trueblood, William R. Holmes, Adam C. Seegmiller, Jonathan Douds, Margaret Compton, Eszter Szentirmai, Megan Woodruff, Wenrui Huang, Charles Stratton, and Quentin Eichbaum. The Impact of Speed and Bias on the Cognitive Processes of Experts and Novices in Medical Image Decision-Making. *Cognitive Research: Principles and Implications*, 3(1):1–14, 2018.

Agnieszka Truszkowska, Brandon Behring, Jalil Hasanyan, Lorenzo Zino, Sachit Butail, Emanuele Caroppo, Zhong-Ping Jiang, Alessandro Rizzo, and Maurizio Porfiri. High-resolution Agent-Based Modeling of COVID-19 Spreading in a Small Town. *Advanced Theory and Simulations*, 4(3), 2021.

United Nations. THE 17 GOALS — Sustainable Development. Technical report, 2021. URL https://sdgs.un.org/goals.

United Nations Office on Drugs and Crime. Global Report on Trafficking in Persons 2016. Technical report. URL https://www.unodc.org/documents/data-and-analysis/glotip/2016_Global_Report_on_Trafficking_in_Persons.pdf.

Gareth Walley, Aditi Shenvi, Peter Strong, and Katarzyna Kobalczyk. cegpy: Modelling with Chain Event Graphs in Python, 2022. URL https://arxiv.org/abs/2211.11366.

Sewall Wright. Correlation and Causation. *Journal of Agricultural Research*, 7(7): 557–585, 1921.

Xuewen Yu and Jim Q. Smith. Causal Algebras on Chain Event Graphs with Informed Missingness for System Failure. *Entropy*, 23(10), 2021. ISSN 1099-4300. doi: 10.3390/e23101308. URL https://www.mdpi.com/1099-4300/23/10/1308.