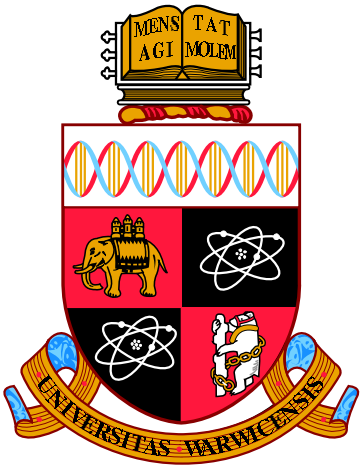


Non-Stratified Chain Event Graphs:
Dynamic Variants, Inference and Applications

by

Aditi Shenvi

Thesis

Submitted to the University of Warwick

for the degree of

Doctor of Philosophy

Mathematics for Real-World Systems CDT

May 2021

WARWICK
THE UNIVERSITY OF WARWICK

Contents

List of Tables iv

List of Figures v

Acknowledgments viii

Declarations x

Abstract xii

Abbreviations xiii

Chapter 1 Introduction 1

1.1 Motivation 1

1.2 Thesis Outline 6

Chapter 2 Graphical Models and Other Preliminaries 9

2.1 An Overview of PGMs 9

2.2 Preliminaries 11

2.2.1 Graph Theory 11

2.2.2 Conditional Independence 14

2.3 Bayesian Networks 16

2.3.1 Dynamic Variants of Bayesian Networks 17

2.3.2 Limitations of Bayesian Networks 18

2.4 Chain Event Graphs 19

2.5 Alternative Graphical Models 22

Chapter 3 Chain Event Graphs 25

3.1 Introduction to CEGs 25

3.1.1 Notation and Semantics 27

3.1.2 Why not just Staged Trees? . . . 33
3.1.3 Stratified CEGs . . . 34
3.2 Conjugate Learning and Model Selection . . . 36
3.2.1 Conjugate Learning . . . 36
3.2.2 Prior Specification . . . 40
3.2.3 Model Selection . . . 42
3.3 Probability Propagation . . . 44
3.4 Dynamic Variants of CEGs . . . 50

Chapter 4 Non-Stratified Chain Event Graphs **54**
4.1 Motivation and Introduction . . . 54
4.2 Non-Stratified CEGs . . . 58
4.3 Construction of Non-Stratified CEGs . . . 60
4.3.1 Motivating the Construction Algorithm . . . 60
4.3.2 CEG Construction Algorithm . . . 63
4.3.3 Related Work . . . 69
4.3.4 Experiments for the CEG Construction Algorithm . . . 69
4.4 Model Selection for Non-Stratified CEGs . . . 71
4.5 Application of the Falls Intervention . . . 73
4.6 Conclusion . . . 78

Chapter 5 Continuous Time DCEGs **80**
5.1 Introduction . . . 81
5.2 Continuous Time Dynamic Chain Event Graphs . . . 83
5.2.1 Incorporating Time-Invariant Covariates . . . 87
5.2.2 Comparison with Existing Models . . . 89
5.3 Conjugate Learning and Model Selection . . . 93
5.3.1 Conjugate Learning . . . 93
5.3.2 Prior Specification . . . 96
5.3.3 Model Selection . . . 97
5.4 A Semi-Markov Representation . . . 100
5.5 Unrolling a CT-DCEG . . . 105
5.6 Probability Propagation in CT-DCEGs . . . 111
5.6.1 A Dynamic Propagation Scheme . . . 112
5.6.2 Propagation Through the Current Model . . . 113
5.6.3 Backward Smoothing . . . 124
5.6.4 Forecasting . . . 124
5.7 Application of the Dynamic Falls Intervention . . . 128

5.8 Conclusion . . . 135

Chapter 6 Bayesian Modelling of Criminal Collaborations with CEGs 138

6.1 Motivation and Introduction . . . 140
6.2 The Reduced Dynamic Chain Event Graph . . . 142
6.2.1 Model Description . . . 144
6.2.2 Implications on Missingness . . . 145
6.3 Review of the Radicalisation and Violent Extremism Model . . . 149
6.3.1 Recurrences in the RVE Model . . . 152
6.4 Related Research . . . 154
6.5 The Dynamic Network Model . . . 155
6.5.1 Pairwise Communications Data . . . 155
6.5.2 Notation . . . 156
6.5.3 The Steady Model . . . 160
6.6 Integrating Decision Support System . . . 166
6.6.1 The Decoupling Methodology . . . 167
6.6.2 IDSS of the Criminal Collaboration Model . . . 170
6.6.3 Cell-Level Threat Scores . . . 172
6.7 Review of a Simulated Example . . . 175
6.7.1 Individual RVEs . . . 176
6.7.2 Network Model . . . 177
6.7.3 Cell-Level RVE Model and Threat Scores . . . 181
6.8 Conclusion . . . 182

Chapter 7 Discussion 186

7.1 Summary of the Contributions of this Thesis . . . 186
7.2 Ongoing Work . . . 188
7.2.1 Mixture Modelling Approach to Model Selection . . . 188
7.2.2 CEG Software . . . 191
7.3 Future Work . . . 192

Appendix A Probability Distributions 194

Appendix B Infinite Tree Probability Measure 197

Bibliography 198

List of Tables

3.1 Interpretation of the Bayes Factor. 39

4.1 Comparison of the baseline algorithm and the optimal time algorithm. . . . 70
4.2 Self-reported health status of individuals aged over 65 living in England and Wales separated by type of usual residence. 74

5.1 The mean posterior transition probabilities and the corresponding edge labels in the CT-DCEG for the vertices in $C^{\mathscr{E}}_{P(2)}$. 126
5.2 The Weibull shape parameter and the posterior Inverse-Gamma distribution for the Weibull scale parameter in the CT-DCEG corresponding to the edges in $C^{\mathscr{E}}_{P(2)}$. 127
5.3 Path probabilities for path λ_i ($i = 1, 2, 3, 4$) in the current model: $p(\lambda_i)$ before propagating the intrinsic and temporal evidence; $p^*(\lambda_i)$ after propagating intrinsic evidence; $\hat{p}(\lambda_i)$ after propagating both the intrinsic and temporal evidence. 128
5.4 $H(e_{i,j})$ refers to the holding time along edge e_{ij} from situations s_i to s_j in Figure 5.14. 130

6.1 Simulated weekly sum of communication duration data. All the zeros in this table indicate that the pair did not communicate through mobile phone call in that week. 179
6.2 Evolution of the prior and posterior parameters for ϕ_{i,j,t_k} during the ten weeks from t_1 to t_{10}. 179
6.3 Cell-level threat measures obtained through the criminal collaboration model and the cell-level RVE. The threat scores $\varphi_C(i)$ are defined as described in Section 6.6.3. 182

List of Figures

1.1 Event trees for the infection process in Example 1.1 as described by (a) case 1, (b) case 2, (c) case 3, and (d) case 4. 4

2.1 A directed, connected, multigraph (a) along with its connected induced subgraph (b) and disconnected subgraph (c). 13
2.2 Graphs (a) and (b) are structurally isomorphic. Graph (c) is obtained from graph (a) by contracting vertices 3 and 5. 14
2.3 Event tree for the infection process described in Example 2.20. 20
2.4 Case 1 encoded within (a) a CEG and (b) a BN. 21
2.5 Case 2 encoded within (a) a CEG, (b) a Bayesian multinet, and (c) a similarity network. 22
2.6 A CEG representing case 3. 23

3.1 Event tree for the infection process described in Example 3.1. 28
3.2 Event tree where edge labels are not fixed *a priori*. (a) and (b) show two possible sets of edge labels. 29
3.3 Staged tree for the infection example. 31
3.4 CEG for the infection example. 33
3.5 Default hyperparameter setting: the numbers represent the units of the imaginary sample arriving at the vertices (inside the circles) and passing along the edges (above the directed edges). 42
3.6 The $\mathscr{E}$-reduced graph of the CEG for the evidence described in Example 3.17. 48

4.1 Event tree for the falls intervention. 56
4.2 (a) Hypothesised BN for the falls intervention; (b) CPT for variable X_{Re}. . . 57
4.3 Event tree for the epilepsy example. 59
4.4 Staged tree for the infection testing example. 61
4.5 First step of the backward iteration. 62
4.6 Second step of the backward iteration. 62

4.7 Third step of the backward iteration. 63
4.8 Staged tree representing the data generating model. 75
4.9 MAP CEG returned by the AHC algorithm. 76
4.10 (a) Original BN returned using the Hill-Climbing algorithm; (b) Best-fitting BN which admits the total order of $X_A \prec X_{Ri} \prec X_T \prec X_F$. 77
4.11 The number of stages in the MAP CEG model of the falls intervention for varying values of $\bar{\boldsymbol{\alpha}}_0$. 77

5.1 Event tree representing the longitudinal infection process. 84
5.2 Hypothesised subgraph of the CT-DCEG for a subgroup of epileptic patients who do not benefit from the anti-epileptic drug being studied. 87
5.3 Graph of the CT-DCEG for the infection process. 89
5.4 The subtrees show the two repeating subtrees for the infection process. . . . 98
5.5 State transition diagram of the SMP for the CT-DCEG of the infection process. 102
5.6 Path from vertex v_i to vertex v_j via vertex v_k. 103
5.7 Unrolled CT-DCEG from passage-slices 1 to 3, i.e. $C_{P(1:3)}$. 106
5.8 $C_{P(1:2)}$ for the infection process. The shaded edges represent hospitalisations. 107
5.9 (a) A CT-DCEG; (b) CEG $C_{P(1:2)}$ of the unrolled CT-DCEG. 108
5.10 Graph of the CT-DCEG in Figure 5.9(a) unrolled from passage-slices k to $k+1$. 109
5.11 (a) A CT-DCEG; (b) graph of CT-DCEG unrolled from passage-slices k to $k+1$; (c) minimal representation of this graph. 110
5.12 $\mathscr{E}$-reduced graphs (a) of current model, (b) past model, and (c) future model for propagating $\mathscr{E}$ and $\mathscr{T}$ as described in Example 5.25. 125
5.13 (a) Calculation of the potentials and emphases in the backward step of our propagation algorithm; (b) The updated current model with the revised transition probabilities obtained through the forward step of our propagation algorithm. 127
5.14 Event tree describing the dynamic falls intervention for assessed individuals in the community. 129
5.15 (a) The data generating CT-DCEG for the simulated dynamic falls intervention; (b) an alternative visualisation of the same CT-DCEG. 132
5.16 The average number of situation clusters (a) and the corresponding average KL divergence (c) for varying values of the imaginary sample size. The average number of edge clusters (b) and the corresponding average KL divergence (d) for varying values of the pseudo-holding time and a fixed imaginary sample size equal to four. 134

6.1 In this figure, C marks the whole underlying population, B marks the general subpopulation of C with the properties that define the individuals the process being studied applies to, and A marks the subpopulation of B that, given the opportunity, would choose to or be chosen to engage with the process. 146
6.2 Graph of the RDCEG for the murder plot. 150
6.3 Structure of network at times t, $t+1$, $t+2$ and $t+3$. 159
6.4 Graphical representation of the 2-time-slice DBN: (a) on a univariate level; (b) on a multivariate level (labelled as graph $\mathcal{G}$). 161
6.5 Overview of the dynamic network model. 167
6.6 The DAG associated with the MDM in Example 6.12. 169
6.7 The DAG associated with the criminal collaboration model. 170
6.8 In both figures, the vertex labels include the prior state probability and edge labels denote the conditional transition probability at time t_1. 176
6.9 Activity data for the four suspects over the observed time period of ten weeks. 177
6.10 Posterior threat state probabilities from the RVE models of the suspects over the ten weeks. 178
6.11 Evolution of ϕ_{i,j,t_k} from time t_3 to t_{10} represented through their posterior densities. 180
6.12 Posterior threat state probabilities from the cell-level RVE model over the ten weeks. 181

Acknowledgments

My first and biggest thanks for this thesis goes to my supervisor Jim Q. Smith. Without his unwavering support, patience and confidence in me, this thesis would not have been possible. I am particularly grateful to Jim for always being very generous with his time for his PhD students.

I am grateful to Silvia Liverani, Oliver Bunnin, Alan Wilson, Jane Hutton, Sandra Eldridge and Laura Bonnett for their guidance and their interest in my work. A special thanks to the Centre for Complexity Science and the MathSys CDT for always having its students' interests and well-being at heart. Particular thanks to Jade Perkins, Debbie Walker, Heather Robson, Colm Connaughton, Magnus Richardson and Yulia Timofeeva. Thanks to all the wonderful and friendly MSc and PhD students at MathSys for being fellow travellers on my PhD journey. I am also grateful to the University of Warwick for funding my research with the Chancellor's scholarship.

Big thanks to Annika, Peter B., Emma, Iliana and Maryam for being amazing friends and for all the runs, climbs and board games! Another big thanks to Rachel for our complexitea sessions and for helping me navigate the CEG world. Thanks to the D2.17 gang – Ayman, Bhavan, KB and Jon – for being my co-conspirators in mischief and procrastination. Special thanks to Sarah-Maëva for believing in me right from the start! Thanks to Nadia, Jev and Avishkar for their uncomplicated friendships. Also thanks to all the lovely people at the complexity flat for making it a homely place for two years. Shoutout to Alicia, Alex and Risa who kept me sane with coffee and climbing while in London. Enormous thanks to Rachel and Peter S. for reading through drafts of my thesis.

Thanks to my friends from back at home – Vaibhavi, Joann, Yash, Hridaya, Siddhant, Mahadevan and Sheetal – for being more like family than friends.

A big heartfelt thank you to my mum, Neela for her unfaltering support, countless sacrifices and selfless love, and to Aparna for being a loving and protective big sister all through my life! Thanks again to both for being my pillars of strength. Thanks to Vivek for being his jovial self and always knowing how to put everyone at ease. Thanks to Vinnu, Kunta and Papa who are dearly missed and without whom, I would not be here today. Thanks to the Bhobe, Shenvi, Pikle, Pothula and Kamath families for their love and support. Thank you to my wonderful partner Gareth for being supportive, patient, putting up with my endless venting, and being equipped with wise words and chocolates, especially in the last few months of writing-up. Finally, this acknowledgement would be incomplete without thanking my lovely doggos, Loki and Laika, whose adorable faces, soggy kisses, musical complaints, and wagging tails make life worth living.

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 4 is based on the material published in two papers for both of which I am the lead author. The first paper, titled *Modelling with Non-Stratified Chain Event Graphs*, appeared in the proceedings of the Bayesian Young Statisticians Meeting. It was the result of joint work with Jim Q. Smith, Robert Walton and Sandra Eldridge. It is cited throughout this thesis as Shenvi et al. (2018). The second, titled *Constructing a Chain Event Graph from a Staged Tree*, appeared in the proceedings of the Tenth International Conference on Probabilistic Graphical Models. It was result of joint work with Jim Q. Smith, and it is cited throughout this thesis as Shenvi and Smith (2020a).

Chapter 5 is partially based on methodologies presented in two pre-prints resulting from joint work with Jim Q. Smith. The first is titled *A Bayesian Dynamic Graphical Model for Recurrent Events in Public Health* and cited as Shenvi and Smith (2019). The second is titled *Propagation for Dynamic Continuous Time Chain Event Graphs* and cited as Shenvi and Smith (2020b). I am the lead author for both of these pre-prints, and they are both currently being revised for submission.

The RDCEG model presented in Chapter 6 was first presented in the pre-print Smith and Shenvi (2018), titled *Assault Crime Dynamic Chain Event Graphs* and later developed in Shenvi and Smith (2019). The work presented in Smith and Shenvi (2018) was led by Jim Q. Smith and I am the second author of this pre-print. The criminal collaboration model presented in Chapter 6 is part of a larger project at the Alan Turing Institute under their

Defence and Security programme. This work has been reported in a pre-print titled *Network Modelling of Criminal Collaborations with Dynamic Bayesian Steady Evolutions*, cited as Bunnin et al. (2020), and was in collaboration with F. Oliver Bunnin and Jim Q. Smith. The material presented in this thesis is my own work, with the following exceptions: Section 6.7 is the work of Bunnin, and is presented in this thesis, with permission, for completeness; Sections 6.4, 6.5.1, 6.5.2 and 6.6.3 were jointly developed with the co-authors. This pre-print is currently being revised for submission and I am the lead author of this work in the revised submission.

The ideas presented in Section 7.2.1, in collaboration with Silvia Liverani, are currently being prepared for submission. This work, along with the work presented in the pre-prints Shenvi and Smith (2020b) and Bunnin et al. (2020) were partially carried out during my PhD Enrichment placement at the Alan Turing Institute.

Abstract

A chain event graph (CEG) is a graphical model that is constructed by identifying the probabilistic symmetries within the tree-based description of a process. CEGs generalise Bayesian networks (BNs) by representing context-specific conditional independencies within their graph topologies. The CEG literature, through the *stratified* CEG class, has demonstrated efficacy over BNs in modelling processes with contextual independence structures.

CEGs are also suited to modelling 'asymmetric' processes with event spaces that do not admit a product space structure. While such processes are common in many domains, they are not easily and effectively modelled by BNs and other graphical models with variable-based topologies. This thesis presents the first exposition of the theory and applications of the more general *non-stratified* CEG class that models asymmetric processes. We demonstrate, through modelling of an asymmetric public health intervention, that the CEG provides a superior representation than the BN in non-product space settings.

We then present a novel dynamic variant of CEGs called the *continuous time dynamic CEG* which has an approximate semi-Markov process representation. We show that this dynamic class generalises and vastly expands the existing subclass of extended dynamic CEGs, first studied in Barclay et al. (2015). We develop semantics unique to this class and propose a dynamic inference scheme for it together with a novel continuous time probability propagation algorithm. In doing this, we are able to utilise any observed information about the temporal evolution of the process to update our beliefs.

Finally, we demonstrate by modelling the evolution of criminal collaborations how the Bayesian paradigm allows us to combine a dynamic CEG model with other disparate models – after due consideration of the independencies between these models – where each model is a component describing a distinct aspect of a complex longitudinal process.

Abbreviations

AHC	Agglomerative hierarchical clustering
APFA	Acyclic probabilistic finite automata
BD	Bayesian Dirichlet score
BN	Bayesian network
CEG	Chain event graph
CHDS	Christchurch health and development study
CIT	Conditional independence tree
CPT	Conditional probability table
CTBN	Continuous time Bayesian network
CT-DCEG	Continuous time dynamic chain event graph
DAG	Directed acyclic graph
DBN	Dynamic Bayesian network
DCEG	Dynamic chain event graph
DLM	Dynamic linear model
FRAT	Falls risk assessment tool
IDSS	Integrating decision support system
KL	Kullback-Leibler
MAP	*Maximum a posteriori*
MAR	Missing at random
MCAR	Missing completely at random
MDM	Multiregression dynamic model
MNAR	Missing not at random
PDG	Probabilistic decision graph

PGM	Probabilistic graphical model
POI	Persons of interest
RDCEG	Reduced dynamic chain event graph
RVE	Radicalisation and violent extremism
SMP	Semi-Markov process
SNA	Statistical network analysis
TNBN	Temporal nodes Bayesian network

Chapter 1

Introduction

1.1 Motivation

A probabilistic graphical model (PGM) is composed of a statistical model and a graph representing the independence relationships between the defining random variables or events of the underlying statistical model. In this way, PGMs bring together two very different branches of mathematics: probability theory and graph theory. The graph of a PGM provides a visually compelling and intuitively intelligible representation of the probabilistic associations encoded within its statistical model. The fact that the essence of these graphs can be typically understood by even those with very little mathematical and statistical training makes PGMs a very useful tool for communication between a statistician and a variety of related vested parties such as the domain experts informing the modelling of the process, other researchers with diverse educational backgrounds, and other stakeholders. Additionally, for the statistician – depending on the family of PGMs chosen and the methodology developed for it – it may be possible to gain a deeper understanding of the process being modelled by simply examining the topology of its graph without making any reference to the parameters of the underlying statistical model.

Within the world of graphical models, the family of Bayesian networks (BNs) (see e.g. Dean and Kanazawa (1989), Cowell et al. (1999), Nodelman et al. (2002), Pearl (2009), and Korb and Nicholson (2010)) have thus far enjoyed tremendous popularity. They have been applied to a wide range of domains including medicine, public health, financial markets, risk analysis, reliability engineering, ecology, meteorology, agriculture, policing, cyber security and forensic analysis.

Notwithstanding the great success of BNs, they do have some shortcomings. In particular, BNs are unable to fully describe asymmetric processes, i.e. processes with event spaces that do not admit a product space structure. We first clarify, with some examples,

the type of asymmetries that can exist within a process. Within this thesis, we consider the following two main types of asymmetries:

- **Asymmetric Independence:** This refers to the presence of context-specific conditional independencies which are independence relationships that hold only for certain values of the conditioning variables, i.e. $X \perp\!\!\!\perp Y|Z = z_1$ but $X \not\perp\!\!\!\perp Y|Z = z_2$ for some variables X, Y and Z where $\perp\!\!\!\perp$ stands for probabilistic independence and the vertical bar shows conditioning variables on the right.
- **Asymmetric Structure:** This refers to the presence of structural missing values, i.e. values that are missing which have no underlying meaningful value, and structural zeros, i.e. observations of a zero frequency for a category of a categorical variable where a non-zero observation is logically restricted.

The second type of asymmetry results in a process whose event space does not admit a product space structure. In this thesis, when we refer to a process being asymmetric, we are referring to the presence of asymmetric structures within the process. To avoid ambiguity, we refer to asymmetric independencies as context-specific or contextual conditional independencies. Asymmetric processes may or may not also exhibit context-specific conditional independencies.

Example 1.1 (Infection example)**.** *Here we consider a simplified topical example. We consider infection in individuals by one of two strains of a certain virus circulating in the population. Infected individuals can get one of two available treatments. The outcome of a treatment, in most cases, is "recovery" and in others is "death". Consider the following cases:*

Case 1: *Given that an individual received treatment 1, their probability of recovery is independent of the strain of the virus by which they were infected. We may represent this information by colouring vertices s_3 and s_5 with the same colour in Figure 1.1(a). A similar relationship holds for an individual who received treatment 2 which we represent by colouring vertices s_4 and s_6 with the same colour in Figure 1.1(a). These two statements can be described succinctly by stating that given the treatment received by an individual, their probability of recovery is independent of the strain of the virus by which they were infected. This process, described by the event tree in Figure 1.1(a), is symmetric in its structure and independence relationships.*

Case 2: *The probability of recovery is independent of the strain of the virus an individual was infected by, given that they received treatment 1 but not if they received treatment 2. Here, the conditional independence relationship holds when the treatment administered to the patient was treatment 1 but not when it was treatment 2. This is an*

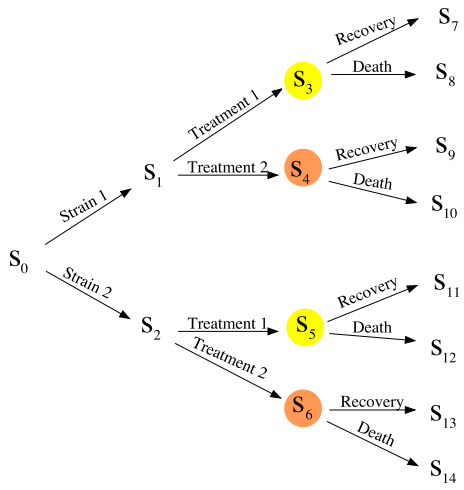
instance of context-specific conditional independence. This process, described by the event tree in Figure 1.1(b), is symmetric in its structure but not in its independence relationships.

Case 3: *Suppose that research showed that the available treatments are ineffective against an infection caused by strain 2 of the virus. Due to this, suppose that no treatment is administered to individuals infected by strain 2. Thus, the variable of treatment does not meaningfully apply to individuals infected by strain 2 of the virus and hence, the treatment variable is structurally missing for these individuals. This process is structurally asymmetric and can be described by the event tree in Figure 1.1(c).*

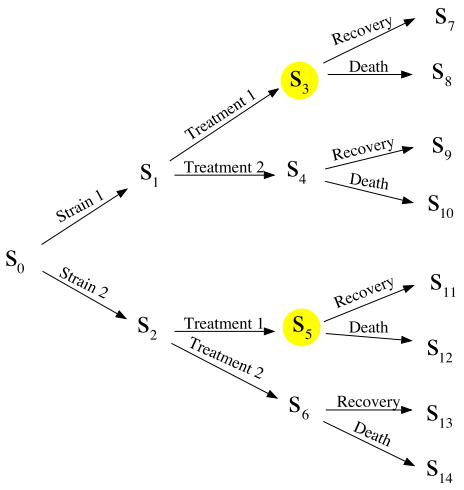
Case 4: *Suppose instead that research showed that treatment 1 is effective against strain 2 of the virus but treatment 1 is ineffective against it. In this case, all those infected with strain 2 of the virus always receive treatment 1. Thus, irrespective of our sample size, we will never observe any individuals who have been infected by strain 2 of the virus and have received treatment 2. Thus, recording zero individuals who have been infected by strain 2 of the virus and have received treatment 2 is a structural zero. Here the state space of the treatment variable for individuals infected by strain 2 is* {"*Treatment 1*"}*. This process is structurally asymmetric and can be described by the event tree in Figure 1.1(d).*

While the conditional independencies encoded with the statistical model of a BN can be inferred by simply interrogating the topology of its graph using the d-separation theorem (e.g. Verma and Pearl (1988), Geiger et al. (1990), and Cowell et al. (1999)), BNs in their unmodified form are unable to express within their graphs context-specific conditional independencies. Uncovering these context-specific conditional independencies requires serious modifications (typically involving trees in some form) to the standard representation and/or inferential process of a BN, see e.g. Boutilier et al. (1996), N. L. Zhang and Poole (1999), and Jabbari et al. (2018). These modifications are discussed in greater detail in Chapter 2. On the other hand, BNs cannot graphically represent structural asymmetries. They are primarily stymied in this respect as they force the process description on a set of variables that are defined *a priori*. Through its model construction with these pre-defined variables, a BN model assumes a symmetric event space that conforms to a product space structure. While structural zeros are hidden away within the conditional probability tables of a BN, there exists no way to accommodate structural missing values within the underlying model of a BN.

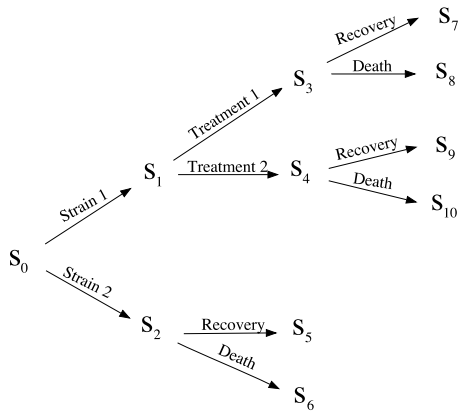
With these limitations in mind, a new family of PGMs called chain event graphs (CEGs) was introduced in Smith and Anderson (2008), developed specifically for processes with asymmetric event spaces and context-specific independencies. The construction of a CEG begins with the elicitation of an event tree describing the process being modelled. Event trees, while being a naturally intuitive framework for describing a process through a sequential unfolding of events (Shafer, 1996), can quickly get unwieldy as the process



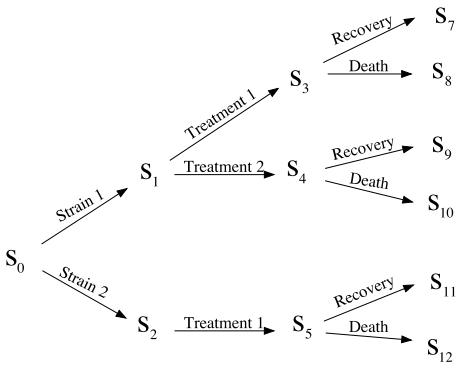
(a)



(b)



(c)



(d)

Figure 1.1: Event trees for the infection process in Example 1.1 as described by (a) case 1, (b) case 2, (c) case 3, and (d) case 4.

being modelled gets larger. A CEG provides a more compact representation of an event tree by exploiting the probabilistic symmetries existing within it. Through their event tree construction, CEGs are able to graphically represent context-specific conditional independencies as well as the atoms of an asymmetric event space.

It has been shown, for example in Smith and Anderson (2008), that CEGs contain the class of discrete BNs as a special case. Over the last decade, several methodological developments have been made for the CEG family: model selection algorithms (Freeman & Smith, 2011a; Silander & Leong, 2013; Cowell & Smith, 2014; Collazo & Smith, 2016), a probability propagation algorithm (Thwaites et al., 2008), a d-separation theorem (Wilkerson, 2020), analysis of missingness through CEGs (Barclay et al., 2014), causal inference (Thwaites et al., 2010; Thwaites, 2013), diagnostics in a CEG (Wilkerson, 2020) and development of dynamic variants of CEGs (Barclay et al., 2015; Collazo, 2017). Applications of CEGs in health studies (Barclay et al., 2013, 2014), educational studies (Freeman & Smith, 2011b) and radicalisation (Collazo, 2017) have also been explored. Chapter 3 formally presents the CEG family and reviews some of the above methodological developments.

As noted above, unlike BNs, CEGs are capable of handling processes with asymmetric event spaces and can graphically represent context-specific conditional independencies. CEGs dealing with processes with symmetric event spaces are called *stratified CEGs*, whereas those dealing with processes with asymmetric event spaces are called *non-stratified CEGs*. In order to demonstrate the efficacy of CEGs over BNs in expressing context-specific independencies, the primary focus of the CEG research thus far has been on developing methodologies and exploring applications for the stratified class. This is because the stratified class, similar to a BN, models processes with event spaces that admit a product space structure. Surprisingly little has been said about what issues result in a process having an asymmetric event space. Thus far, the applicability of existing CEG methodologies to the non-stratified class has not been studied, nor has there been an application with a non-stratified CEG that has systematically analysed a real-world process.

We noted above that some dynamic variants of CEGs have been developed. With the exception of the extended dynamic CEG (Barclay et al., 2015), the other dynamic CEGs (DCEGs) (Barclay et al., 2015; Collazo, 2017) have been developed for longitudinal processes evolving in discrete time. Further, these discrete time DCEGs have been developed for processes satisfying the Markov property. Hence, information about the time spent by an individual at the various states depicted by such a DCEG do not offer any discriminatory information about the process, and so, the temporal evolution of these processes is typically disregarded. On the other hand, extended DCEGs take into account such conditional holding times at the various states depending on the state occupied next. By explicitly accounting for the holding times, we are able to model longitudinal processes where observations

are recorded as the events occur rather than at regular time intervals.

However, on further inspection, we find that the extended DCEGs represent a special subclass of the general class of DCEGs evolving in continuous time. In particular, extended DCEGs do not directly investigate when the conditional holding times associated with different transitions follow the same distribution. Instead, they assume that the conditional holding time distributions for two states are the same whenever the conditional transition probabilities out of the two states are equivalent. Further, extended DCEGs are described as an extension of discrete time DCEGs and hence, they do not have the necessary range of semantics to appropriately describe a process evolving in continuous time.

Finally, we note that the applications of CEGs (DCEGs) considered thus far are those where the whole process can be described by the CEG (DCEG). Within the Bayesian paradigm, we are able to systematically combine together disparate models – after due consideration of the independencies between these models – where each model is a component of a larger composite model describing a distinct aspect of a complex longitudinal process. Thus a CEG (DCEG) can be one such model within a larger modelling framework.

With the above points in mind, this thesis aims to address the following research questions:

1. What are the underlying issues that result in a process having an asymmetric event space? Are there any differences in constructing a non-stratified CEG from its event tree? Are the model selection methodologies developed for stratified CEGs also applicable to the non-stratified class? (Chapter 4)
2. How can we define a class of continuous time DCEGs with the necessary semantics such that it can describe a longitudinal complex process, evolving in continuous time, with its different components evolving at different rates? How does inference work in this class? How can we address model selection within this class as it contains a very large search space? (Chapter 5)
3. How can a continuous time DCEG be combined with other models, each describing a distinct part of a complex process? Under what conditions can we combine these models such that we can still estimate the parameters of each model independently? (Chapter 6)

1.2 Thesis Outline

The rest of this thesis is organised as follows.

In Chapter 2 we begin by reviewing the relevant preliminary and graph theoretical concepts. We then review BNs, their two important dynamic variants: discrete time dy-

namic BNs and continuous time BNs, and discuss the limitations of the BN family with reference to modelling processes with context-specific conditional independencies and asymmetric event spaces. We conclude this chapter with a discussion of alternative PGMs for asymmetric processes.

In Chapter 3 we review CEGs. This review is primarily based on the stratified class of CEGs as the methodologies developed thus far have generally been customised to this class. This review considers conjugate updating of the parameters of a CEG model, model selection algorithms for the stratified class, and the CEG probability propagation algorithm. We conclude this chapter by discussing the relevant existing dynamic variants of CEGs, namely discrete time DCEGs and the continuous time extended DCEGs.

Chapter 4 begins with a motivation and introduction to what type of issues lead to processes having asymmetric event spaces, and thereon, discusses the real-world relevance of the non-stratified CEG class. We prove that a CEG is uniquely defined by its staged tree and present a backward iterative construction algorithm with an optimal stopping criterion to transform any staged tree (stratified or non-stratified) into a CEG – which has been missing from the existing literature. We next describe how the model selection methodologies developed for stratified CEGs can be extended to the non-stratified class, and present an application of the non-stratified CEG class on an intervention to reduced falls-related injuries among the elderly.

Chapter 5 introduces the general class of DCEGs evolving in continuous time called the continuous time DCEG (CT-DCEG), of which the extended DCEGs are a special subclass. Along with introducing new semantics customised to this continuous time setting, we demonstrate how time-invariant covariates (e.g. age, socioeconomic background, chronic health conditions) which do not have any associated holding times can be incorporated within a CT-DCEG. We show that all CT-DCEGs enjoy an alternative, possibly approximate, representation as a semi-Markov process. Within this chapter, we also explore conjugate learning, model selection and a dynamic propagation scheme for this class. We then present an application of the CT-DCEG on a dynamic extension of the falls intervention.

Chapter 6 illustrates how a member of the CEG family can be combined with other models within a wider modelling framework to describe a complex multi-faceted process. We model the evolution of criminal collaborations among suspected criminals. This criminal collaboration model consists of two parts. The first is an existing lone criminal model introduced in Smith and Shenvi (2018) and developed in (Bunnin & Smith, 2019). This model utilises a new subclass of CT-DCEGs called the reduced DCEG which we formally introduce for the first time in this thesis. We then review the lone criminal model. Next, we introduce the second part of the criminal collaboration model which is a dynamic weighted network model. Finally, we demonstrate how these two parts can be combined together, and

illustrate how this composite model can be used to create bespoke cell-level threat scores that indicate the imminence of the threat posed by a known or suspected criminal cell. To our knowledge this is the first example where a DCEG/CEG is used as one of several components within a composite model.

We conclude with Chapter 7 which presents a summary of the contributions of this thesis, and a discussion of ongoing and future work.

Chapter 2

Graphical Models and Other Preliminaries

We begin this chapter by providing a broad overview of PGMs in Section 2.1, building on the discussion started in Chapter 1. In Section 2.2 we define some preliminary graph theoretic concepts, and discuss the concepts of independence and conditional independence.

Since this thesis focuses on the CEG family whose graphs are directed, it is of relevance to compare this family to the competing family of BNs which are currently the most popular directed graphical models within the literature. We present a brief review of BNs in Section 2.3. Within this section, we also review the relevant dynamic variants of BNs, namely the dynamic BN and continuous time BN. We then discuss the limitations of BNs as well as the approaches proposed in the literature to overcome them. In Section 2.4 we present a simple and non-technical description of CEGs. Here we illustrate through an example how CEGs easily overcome the main limitation of BNs due to their tree-based construction. This section is placed here to enable the reader to compare the CEG – through the example – to the BN framework. A detailed technical review of CEGs is deferred to Chapter 3. Finally, in Section 2.5, we present two directed graphical models that are not associated with the BN family but can be considered as alternatives to the CEG for modelling asymmetric processes.

2.1 An Overview of PGMs

Recall that a PGM is a statistical model with a graphical representation that facilitates interaction between statisticians, domain experts and decision makers. Graphs have a long history of being used to describe statistical models, which began as early as 1921 with the introduction of path analysis by geneticist Sewall Wright (Sewall, 1921). Pearl (2009) states

that the role of graphs in statistical models is given as follows:

1. to provide an intuitive way of expressing substantive assumptions about a process;
2. to facilitate a compact representation of joint probability distributions;
3. to provide a way to perform efficient inference.

Thus, the graph of a PGM not only provides a representational advantage by being a visual interface but it also compactly represents the dependencies among its variables. Further, Pearl (2009) emphasised that PGMs are intended to graphically encode prior judgements of independencies *before* beginning any probabilistic considerations. Pearl (1986) reasons that it is easier for domain experts to make quick and reliable probabilistic judgements about a small number of variables at a time rather than the entire complex system at once. Thus, the compact representation provided by a graph of the independencies among the variables makes it easier to elicit relevant conditional probabilities from the domain expert. Additionally, these independencies enable us to factorise the joint distribution of the variables into several smaller distributions – each involving a smaller subset of variables. Through the computational advantages leveraged by the decomposition of the joint distribution, PGMs facilitate quick and efficient inference and propagation of evidence within a complex high-dimensional system. Further, these computational benefits also enable development of efficient model selection algorithms.

PGMs can be broadly categorised as those represented by undirected graphs, directed graphs and mixed graphs (Studený, 2005). Undirected graphs are also known as Markov networks or Markov fields and they represent conditional independence relationships within the graph with undirected edges (see e.g. Lauritzen (1996)). Directed graphs are typically represented by DAGs. The most popular among these is the BN. We shall discuss BNs, some of its dynamic variants and its limitations in Section 2.3. The third category consists of graphs which contain both directed and undirected edges. These are also known as chain graphs and they were introduced by Lauritzen and Wermuth (1989).

While a majority of the PGMs represent conditional independencies among a set of random variables; there are instances of PGMs which take a more event-based approach. Schlaifer and Raiffa (1961) presented decision trees – although not historically included within the literature of PGMs – to describe the sequential unfolding of events through a combination of controllable (decision) nodes and uncontrollable (chance) nodes leading to the final outcome represented by the leaves (value nodes) of the tree and often associated with a utility or monetary value. The need for a compact representation of decision trees for larger problems led to the development of influence diagrams in Howard and Matheson (1981). Influence diagrams are represented graphically by a DAG composed of decision, chance and value nodes, and in fact, they are a generalisation of BNs (see e.g. Smith

(2010)). However, under their original formulation, influence diagrams were viewed as a "front end" of a decision analysis problem with the evaluation still relying on the decision tree. This changed when Shachter (1986) presented an algorithm for evaluating a decision problem directly through the influence diagram representation. Due to this, decision trees lost attention within the world of graphical modelling. Influence diagrams, on the other hand, remain popular as a tool for complex decision analysis.

While decision trees may have fallen out of use for reasoning and modelling of uncertainty, many processes are still best described by domain experts as an evolution of events. Translating such an event-based tree description to a variable-based BN is not trivial. Particularly when the process exhibits context-specific conditional independencies or asymmetric developments, a BN is unable to fully describe such processes (see Section 2.3.2). This led to the development of the graphical modelling family of CEGs described in Chapter 3. Unlike a BN or indeed, an influence diagram, a CEG retains all the original root-to-leaf paths represented in its corresponding tree description. This thesis focuses on the development of a class of CEGs for processes which do not have an appropriate alternative BN representation. The rest of this chapter aims to present the necessary background for the reader as well as some examples to motivate the need for such a class of CEGs.

2.2 Preliminaries

2.2.1 Graph Theory

In this section we review some graph theoretical concepts that will be used throughout this thesis. For further details on these concepts, see D. B. West (2001).

Definition 2.1 (Graph)**.** *A graph $\mathcal{G} = (V(\mathcal{G}), E(\mathcal{G}))$ consists of a vertex set $V(\mathcal{G})$ and an edge set $E(\mathcal{G})$ such that each edge in $E(\mathcal{G})$ connects a pair of vertices in $V(\mathcal{G})$. The graph $\mathcal{G}$ is said to be finite if both $V(\mathcal{G})$ and $E(\mathcal{G})$ are finite sets; otherwise it is said to be infinite.*

Definition 2.2 (Directed, Undirected and Mixed Graphs)**.** *A graph $\mathcal{G} = (V(\mathcal{G}), E(\mathcal{G}))$ is*

- *directed if each edge in $E(\mathcal{G})$ has an associated directionality or orientation which is represented in the graph by an arrow from the emanating vertex to the terminating vertex;*
- *undirected if each edge in $E(\mathcal{G})$ does not have any directionality which is represented in the graph by a line between the pair of vertices;*
- *mixed if $E(\mathcal{G})$ contains both directed and undirected edges.*

The graph of a CEG is directed. Hence, the definitions below are presented for directed graphs.

Definition 2.3 (Simple Graphs and Multigraphs)**.** *A directed graph $\mathcal{G} = (V(\mathcal{G}), E(\mathcal{G}))$ is said to be a simple graph if it has no directed edges from a vertex to itself (also known as a loop) and has at most one edge of a given directionality between any pair of vertices; whereas it is said to be a multigraph if it is allowed to have loops and multiple edges of a given directionality between a pair of vertices.*

Definition 2.4 (Subgraph and Induced Subgraph)**.** *Given two graphs $\mathcal{G}_1 = (V(\mathcal{G}_1), E(\mathcal{G}_1))$ and $\mathcal{G}_2 = (V(\mathcal{G}_2), E(\mathcal{G}_2))$, the graph $\mathcal{G}_2$ is said to be a subgraph of the graph $\mathcal{G}_1$ if $V(\mathcal{G}_2) \subseteq V(\mathcal{G}_1)$ and $E(\mathcal{G}_2) \subseteq E(\mathcal{G}_1)$. Further, graph $\mathcal{G}_2$ is said to be an induced subgraph of the graph $\mathcal{G}_1$ if every edge in $E(\mathcal{G}_1)$ whose both endpoints are in $V(\mathcal{G}_2)$ is also an edge in $E(\mathcal{G}_2)$.*

Definition 2.5 (Walk, Path and Cycle)**.** *A walk (of length k) in a directed graph $\mathcal{G} = (V(\mathcal{G}), E(\mathcal{G}))$ is a non-empty alternating sequence $v_0e_0v_1e_1 \dots e_{k-1}v_k$ where $v_i \in V(\mathcal{G})$ and $e_j \in E(\mathcal{G})$ such that edge e_j emanates from vertex v_j and terminates in v_{j+1} for $0 \leq j < i \leq k$. A walk where all vertices are distinct is called a path. A walk where $v_0 = v_k$ is called a cycle.*

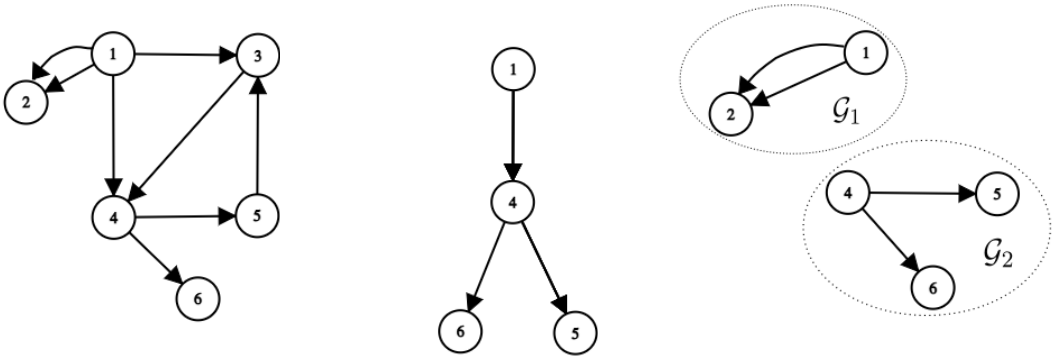
Definition 2.6 (Connected Graph and Connected Components)**.** *A directed graph $\mathcal{G} = (V(\mathcal{G}), E(\mathcal{G}))$ is said to be connected if its undirected version is non-empty and every pair of distinct vertices is connected by a path. If a graph $\mathcal{G}$ is disconnected, each maximally connected subgraph of $\mathcal{G}$ is called a connected component.*

Definition 2.7 (Directed Acyclic Graph)**.** *A directed acyclic graph (DAG) is a directed graph which does not contain any cycles.*

Definition 2.8 (Parent and Child)**.** *In a directed graph $\mathcal{G} = (V(\mathcal{G}), E(\mathcal{G}))$, if there exists an edge in $E(\mathcal{G})$ that emanates from vertex v and terminates in vertex v' for $v, v' \in V(\mathcal{G})$, v is said to be the parent of v', and v' is said to be the child of v.*

Definition 2.9 (Tree)**.** *A directed graph $\mathcal{G} = (V(\mathcal{G}), E(\mathcal{G}))$ is said to be a tree if its undirected version is connected and contains no cycles. A vertex $v_0 \in V(\mathcal{G})$ is designated as the root vertex. The root vertex has no parents. Each other vertex in the tree has exactly one parent. The vertices with no children are called leaves. A disjoint collection of trees is known as a forest.*

Example 2.10. *All three graphs in Figure 2.1 are directed graphs. Figure 2.1(a) shows a multigraph as it contains two edges of the same directionality from vertex 1 to vertex 2 whereas Figure 2.1(b) shows a simple graph which is also a DAG and a tree. Consider the*



(a) (b) (c)

Figure 2.1: A directed, connected, multigraph (a) along with its connected induced subgraph (b) and disconnected subgraph (c).

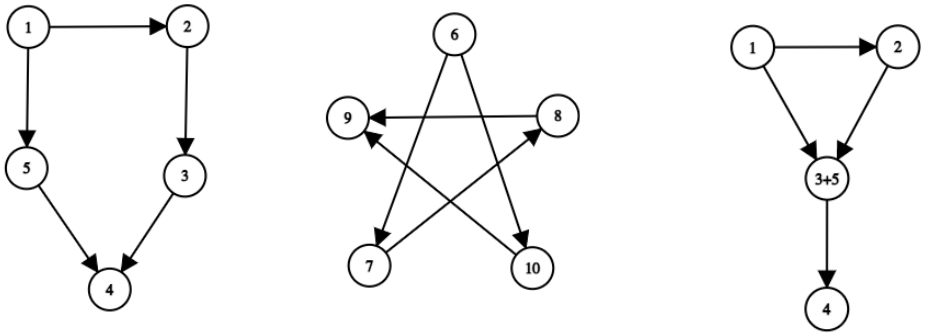
following sequences in the graph in Figure 2.1(a):

$$\rho_1 = v_1, e_{1,4}, v_4, e_{4,5}, v_5, e_{5,3}, v_3, e_{3,4}, v_4;$$
$$\rho_2 = v_1, e_{1,4}, v_4, e_{4,5}, v_5, e_{5,3}, v_3;$$
$$\rho_3 = v_4, e_{4,5}, v_5, e_{5,3}, v_3, e_{3,4}, v_4.$$

Sequence ρ_1 is a walk, ρ_2 is a path and ρ_3 is a cycle. Figure 2.1(b) shows a connected, induced subgraph of the graph in Figure 2.1(a) whereas Figure 2.1(c) shows a disconnected subgraph of the graph in Figure 2.1(a). The subgraphs in Figure 2.1(c) labelled $\mathcal{G}_1$ and $\mathcal{G}_2$ are its connected components.

Definition 2.11 (Vertex Contraction)**.** *In a directed graph $\mathcal{G} = (V(\mathcal{G}), E(\mathcal{G}))$, the contraction of a pair of vertices v and v' results in a graph where v and v' are replaced by a single vertex v^* such that the set of edges terminating at (emanating from) v^* are given by the union of edges terminating at (emanating from) vertices v and v'. Here, vertices v and v' need not be connected by an edge prior to the contraction.*

Definition 2.12 (Graph Isomorphism)**.** *Two directed graphs $\mathcal{G}_1 = (V(\mathcal{G}_1), E(\mathcal{G}_1))$ and $\mathcal{G}_2 = (V(\mathcal{G}_2), E(\mathcal{G}_2))$ are isomorphic if there is a bijection $f : V(\mathcal{G}_1) \to V(\mathcal{G}_2)$ such that there is an edge from vertex v to vertex v' in $E(\mathcal{G}_1)$ for $v, v' \in V(\mathcal{G}_1)$ if and only if there exists an edge from vertex $f(v)$ to vertex $f(v')$ in $E(\mathcal{G}_2)$ for $f(v), f(v') \in V(\mathcal{G}_2)$. Such an isomorphism is adjacency-preserving or structure-preserving. In coloured graphs, isomorphism can also be defined to be colour-preserving.*



(a) (b) (c)

Figure 2.2: Graphs (a) and (b) are structurally isomorphic. Graph (c) is obtained from graph (a) by contracting vertices 3 and 5.

Example 2.13. *Consider the graphs in Figure 2.2. The graphs in Figure 2.2(a) (say $\mathcal{G}_1$) and Figure 2.2(b) (say $\mathcal{G}_2$) are structurally isomorphic where the bijection is given as follows*

$$f : V(\mathcal{G}_1) \to V(\mathcal{G}_2)$$
$$\text{such that } f(i) = i + 5.$$

The graph in Figure 2.2(c) is obtained by contracting vertices 3 and 5 in the graph in Figure 2.2(a) into a single vertex.

2.2.2 Conditional Independence

We begin by presenting a definition of independence and conditional independence below.

Definition 2.14 (Independence and Conditional Independence)**.** *Consider three disjoint subsets $\boldsymbol{X}$, $\boldsymbol{Y}$ and $\boldsymbol{Z}$ of a set of random variables $\boldsymbol{V} = \{V_1, V_2, \ldots, V_n\}$. We say that $\boldsymbol{X}$ and $\boldsymbol{Y}$ are independent if and only if their joint probability density or mass function $p(\boldsymbol{x}, \boldsymbol{y})$ decomposes as follows*

$$p(\boldsymbol{x}, \boldsymbol{y}) = p(\boldsymbol{x})p(\boldsymbol{y}), \tag{2.1}$$

for all values $\boldsymbol{x}$ and $\boldsymbol{y}$ of the random variable sets $\boldsymbol{X}$ and $\boldsymbol{Y}$. Further, $\boldsymbol{X}$ and $\boldsymbol{Y}$ are said to be conditionally independent given $\boldsymbol{Z}$, written as $\boldsymbol{X} \perp\!\!\!\perp \boldsymbol{Y} \,|\, \boldsymbol{Z}$, if and only if

$$p(\boldsymbol{x} \,|\, \boldsymbol{y}, \boldsymbol{z}) = p(\boldsymbol{x} \,|\, \boldsymbol{z}) \quad \text{whenever } p(\boldsymbol{y}, \boldsymbol{z}) > 0, \tag{2.2}$$

for all values $\boldsymbol{x}, \boldsymbol{y}$ *and* $\boldsymbol{z}$ *of the random variable sets* $\boldsymbol{X}$, $\boldsymbol{Y}$ *and* $\boldsymbol{Z}$.

We next explore how conditional independence relationships can be expressed graphically. Dawid (1979) and Spohn (1980) theorised the statistical properties of conditional independence which are given below. Let $\mathbf{W}, \mathbf{X}, \mathbf{Y}$ and $\mathbf{Z}$ be disjoint subsets of a set of random variables $\mathbf{V} = \{V_1, V_2, \ldots, V_n\}$. Let $\cdot \perp\!\!\!\perp \cdot | \cdot$ be the ternary conditional independence relation. The following four properties hold for any underlying probability measure

$$\begin{aligned}
\textbf{Symmetry} &: \mathbf{X} \perp\!\!\!\perp \mathbf{Y} \,|\, \mathbf{Z} \implies \mathbf{Y} \perp\!\!\!\perp \mathbf{X} \,|\, \mathbf{Z};\\
\textbf{Decomposition} &: \mathbf{X} \perp\!\!\!\perp (\mathbf{Y}, \mathbf{W}) \,|\, \mathbf{Z} \implies \mathbf{X} \perp\!\!\!\perp \mathbf{Y} \,|\, \mathbf{Z} \text{ and } \mathbf{X} \perp\!\!\!\perp \mathbf{W} \,|\, \mathbf{Z};\\
\textbf{Weak union} &: \mathbf{X} \perp\!\!\!\perp (\mathbf{Y}, \mathbf{W}) \,|\, \mathbf{Z} \implies \mathbf{X} \perp\!\!\!\perp \mathbf{Y} \,|\, (\mathbf{Z}, \mathbf{W});\\
\textbf{Contraction} &: \mathbf{X} \perp\!\!\!\perp Y \,|\, (\mathbf{Z}, \mathbf{W}) \text{ and } \mathbf{W} \perp\!\!\!\perp \mathbf{Y} \,|\, \mathbf{Z} \implies (\mathbf{W}, \mathbf{X}) \perp\!\!\!\perp \mathbf{Y} \,|\, \mathbf{Z}.
\end{aligned}$$

This axiomatic basis of conditional independence was then linked to vertex separation conditions in graphs by Pearl and Paz (1986). It is through this connection that conditional independence relationships can be represented by separation in graphs. The above four properties constitute the *semi-graphoid axioms*, and any independence model that respects these four properties is called a *semi-graphoid*. Further, if the underlying probability measure is strictly positive, a fifth property holds

$$\textbf{Intersection} : \mathbf{X} \perp\!\!\!\perp \mathbf{Y} \,|\, (\mathbf{Z}, \mathbf{W}) \text{ and } \mathbf{X} \perp\!\!\!\perp \mathbf{W} \,|\, (\mathbf{Y}, \mathbf{Z}) \implies \mathbf{X} \perp\!\!\!\perp (\mathbf{Y}, \mathbf{W}) \,|\, \mathbf{Z}.$$

An independence model that respects these five properties is called a *graphoid*.

We now present the related concept of context-specific conditional independence. We shall explore the representation of context-specific conditional independence in graphical models later in this chapter.

Definition 2.15 (Context-Specific Conditional Independence (Boutilier et al., 1996))**.** *Consider three disjoint subsets* $\boldsymbol{X}$, $\boldsymbol{Y}$ *and* $\boldsymbol{Z}$ *of a set of random variables* $\boldsymbol{V} = \{V_1, V_2, \ldots, V_n\}$. *We say that* $\boldsymbol{X}$ *and* $\boldsymbol{Y}$ *are said to be context-specific conditionally independent given the context* $\boldsymbol{Z} = z$, *written as* $\boldsymbol{X} \perp\!\!\!\perp \boldsymbol{Y} \,|\, \boldsymbol{Z} = z$, *if and only if for some value* z *of* $\boldsymbol{Z}$

$$p(\boldsymbol{x}|\boldsymbol{y}, z) = p(\boldsymbol{x}|z) \quad \textit{whenever } p(\boldsymbol{y}, z) > 0. \tag{2.3}$$

for all values $\boldsymbol{x}$ *and* $\boldsymbol{y}$ *of the random variable sets* $\boldsymbol{X}$ *and* $\boldsymbol{Y}$.

Appendix A provides the forms of the mass and density functions of the probability distributions that feature in this thesis.

2.3 Bayesian Networks

Bayesian networks are a graphical modelling family that represent the probabilistic relationships among a set of variables in terms of conditional independence statements. The BN was first introduced in Pearl (1986) and has since been applied to a wide range of domains for reasoning in the presence of uncertainty. The vertices of a BN represent the variables of interest and a directed edge between two vertices represents informational or causal dependencies between the two variables. These dependencies are quantified by conditional probabilities of a variable given the values assumed by its parent variables in the network. For a detailed review of BNs and their applications, see for example Cowell et al. (1999), Koller and Friedman (2009), Pearl (2009), and Korb and Nicholson (2010).

Definition 2.16 (Bayesian Network)**.** *A Bayesian network (BN) $\mathcal{B} = (\mathcal{G}, P)$ is a probabilistic graphical model over a set of variables $\boldsymbol{X} = \{X_1, X_2, \ldots, X_n\}$. Here $\mathcal{G} = (V(\mathcal{G}), E(\mathcal{G}))$ is a DAG whose vertices are given by the variables in $\boldsymbol{X}$, and P is a joint probability distribution over the variables $\boldsymbol{X}$. The edge set $E(\mathcal{G}) \subseteq V(\mathcal{G}) \times V(\mathcal{G})$ consists of directed arcs such that lack of an edge between two nodes represents conditional independence between the variables represented by the nodes, and similarly, edges between nodes encode conditional dependence. This conditional independence structure allows the joint probability P to be factorised by the chain rule as*

$$P(\boldsymbol{X} = \boldsymbol{x} \,|\, \mathcal{G}) = \prod_{X_i \in \boldsymbol{X}} P(X_i = x_i | Pa(X_i) = x_{Pa(X_i)})$$

where $Pa(X_i)$ is the set of parents of the node X_i in $\mathcal{G}$.

The DAG of the BN encodes the following conditional independence statements

$$X_i \perp\!\!\!\perp Nd(X_i) \backslash Pa(X_i) \,|\, Pa(X_i) \tag{2.4}$$

where $Nd(X_i)$ are the non-descendants of X_i, i.e. all the variables in $\mathcal{G}$ that do not have a directed path from X_i to themselves. This is known as the *local directed Markov property*. However, more conditional independence relationships can be read directly from the graph of a BN using the *d-separation theorem*; first defined by Verma and Pearl (1988) and later presented as the *global directed Markov property* by Lauritzen (1996).

Definition 2.17 (d-Separation Theorem (Verma & Pearl, 1988))**.** *Let $\boldsymbol{X}$, $\boldsymbol{Y}$ and $\boldsymbol{Z}$ be three disjoint subsets of vertices in the DAG $\mathcal{G}$ of a BN $\mathcal{B} = (\mathcal{G}, P)$. We say that $\boldsymbol{Z}$ d-separates $\boldsymbol{X}$ from $\boldsymbol{Y}$ if and only if there is no undirected path (i.e. ignoring the directionality of the edges) in $\mathcal{G}$ from a vertex in $\boldsymbol{X}$ to a vertex in $\boldsymbol{Y}$ along which the following conditions hold:*

1. *Every collider vertex – a vertex with converging arrows – either is in or has a descent in* $\mathbf{Z}$*;*
2. *Every other vertex is outside* $\mathbf{Z}$*.*

Through its factorised representation of the joint probability distribution of a model, the BN exploits the conditional independence relationships among the variables of a process. This allows for a reduction of the dimensionality of a large complex process. Further, through the d-separation theorem, all the conditional independence relationships encoded in the BN model can be read directly from its graph topology. Since its inception, research in BNs has been a very active research field and BNs have been applied to a wide range of domains; thereby establishing themselves as a popular modelling tool. A wide range of BN methodologies now exist for model selection (e.g. Heckerman et al. (1995) and Cussens (2008, 2011)), inference (e.g. Shafer and Shenoy (1990) and Darwiche (2003)) and causal discovery (e.g. Pearl (1994, 2009)). In Section 2.3.2 we discuss some limitations of BNs.

2.3.1 Dynamic Variants of Bayesian Networks

Over the decades, several dynamic variants of BNs have been proposed in the literature, such as dynamic BNs (DBNs) (Dean & Kanazawa, 1989), continuous time BNs (CTBNs) (Nodelman et al., 2002), hybrid time BNs (Liu et al., 2017) and temporal nodes BNs (Arroyo-Figueroa & Sucar, 1999). In this review we shall focus on the two main dynamic variants: DBNs and CTBNs.

Definition 2.18 (Dynamic Bayesian Network)**.** *A dynamic Bayesian network (DBN) is a dynamic variant of the BN that evolves in discrete time. A DBN, defined over a set of variables* $\boldsymbol{X}(t) = \{X_1(t), X_2(t), \ldots, X_n(t)\}$ *representing a time-series, is given by the tuple* $(\mathcal{B}_1, \ldots, \mathcal{B}_n)$ *where* $\mathcal{B}_1$ *is the initial BN over* $\boldsymbol{X}(1)$ *and each subsequent BN* $\mathcal{B}_t$ *represents the state of the system at time-slice* t *over* $\boldsymbol{X}(t)$ *for* $t \geq 2$*. Assuming the system satisfies the first-order Markov property, the BN* $\mathcal{B}_t$ *is connected to the BN* $\mathcal{B}_{t+1}$ *by directed inter-slice temporal arcs to represent the effect of the variables at time* t *on the variables at time* $t + 1$*.*

A common simplification of the DBN is to assume stationarity of the graphical structure and the model parameters over time. Such a DBN is called a 2-time-slice DBN and can be compactly given by the tuple $(\mathcal{B}_1, \mathcal{B}_{\rightarrow})$ where $\mathcal{B}_1$ is the initial BN and $\mathcal{B}_{\rightarrow}$ is the transition BN that describes the dependencies of a variable X at time t given the values of its parents in time-slices t and $t - 1$.

Definition 2.19 (Continuous Time Bayesian Network)**.** *A continuous time Bayesian network (CTBN)* $\mathcal{B} = (\mathcal{B}_0, \mathcal{B}_{\rightarrow})$ *for a set of variables* $\boldsymbol{X} = \{X_1, X_2, \ldots, X_n\}$*, such that each* X_i *has a finite state space, evolves in continuous time and consists of two components. The*

first is the initial distribution P_0 which is specified as a BN $\mathcal{B}_0$ over $\boldsymbol{\mathcal{X}}$. The second is a continuous transition model $\mathcal{B}_{\rightarrow}$ consisting of a possibly cyclic graph $\mathcal{G}$ whose nodes are the variables of $\boldsymbol{\mathcal{X}}$ and where the dependencies between the variables, denoted by directed edges, are quantified by a conditional intensity matrix $Q_{X_i|Pa(X_i)}$ for each variable $X_i \in \boldsymbol{\mathcal{X}}$.

A CTBN models the evolution of a complex system evolving in continuous time. It assumes that the system experiences at most one transition at a time, i.e. no two transitions may occur simultaneously. The conditional intensity matrix for a variable models the transition from one state of the variable to another state. Each variable, conditioned on its parents, is modelled by a continuous time Markov process. Thus, an implicit assumption of the model is that the holding time in any state of a given variable follows an exponential distribution. A detailed discussion on CTBNs is presented in Chapter 5 in Section 5.2.2 where it is compared to the continuous time DCEG model presented therein.

2.3.2 Limitations of Bayesian Networks

Despite their inferential and representational advantages and success across a wide range of domains, BNs are not the optimal choice of model for certain processes. In particular, BNs are not a suitable model for processes that exhibit one or both asymmetries described below:

1. context-specific conditional independence (see Definition 2.15) where the independence relationship holds only for certain values of the conditioning variable;
2. asymmetric event space that does not admit a product space structure.

There have been several modifications suggested in the literature to enable BNs to accommodate context-specific conditional independencies. Most of these modifications are specifically designed for inferential gains whereas some others also consider the representational benefits of graphically expressing these independencies within the BN. In an early attempt, Boutilier et al. (1996) and N. L. Zhang and Poole (1999) replace the conditional probability table (CPT) of the BN with tree structures to represent context-specific conditional independencies. Boutilier et al. (1996) then uses these tree-structured CPTs to rearrange the graph of the BN such that a single variable could be represented by multiple vertices in this modified graph. Poole and Zhang (2003) proposed *contextual belief networks* which are BNs where the probability assignments for each variable are only specified for its parent contexts. Here the parent contexts for a variable are constructed based on the variable being conditionally independent of a subset of its parent variables when conditioned on the realisations of its other parent variables. All the above approaches aim to exploit contextual independencies within a process for faster and more efficient inference,

and not for representational benefits. Similarly, several methods have been proposed for learning BNs in the presence of contextual independencies but they do not offer representational improvements. These search methods usually involve learning the global structure of the BN, followed by learning the local structure of each CPT of the BN given a particular global structure (see e.g. Friedman and Goldszmidt (1996), Chickering et al. (1997), Jabbari et al. (2018), and Shen et al. (2020)).

Geiger and Heckerman (1996) proposed an extension to the BN framework in the form of *Bayesian multinets* to visually represent context-specific conditional independencies within the graphical representation of the BN. In a Bayesian multinet defined over a set of random variables $\boldsymbol{X} = \{X_1, X_2, \ldots, X_n\}$, one of the variables (say X_h, $1 \leq h \leq n$) is selected to be the "hypothesis variable" and the values in the sample space of the hypothesis variable are referred to as "hypotheses". The hypothesis variable is chosen such that a partition of the sample space of X_h can be formed where each set of hypotheses in the partition gives rise to a distinct set of conditional independencies, and hence a distinct BN over the random variables $\boldsymbol{X}$. A Bayesian multinet is a collection of distinct local BNs where each BN represents the conditional independencies among the variables $\boldsymbol{X}$ for a specified subset of hypotheses. In this way, Bayesian multinets encode asymmetric independencies. Another related class of models is the similarity network proposed by Heckerman (1990). A similarity network is also a collection of distinct local BNs where each local BN helps to discriminate among two subsets of hypotheses, say h_1 and h_2, in the partition of the sample space of X_h. A local BN differentiating between sets h_1 and h_2 is defined only over the random variables in $\boldsymbol{X}$ that help to discriminate between these two sets of hypotheses. Both of the above approaches result in a fragmented representation of a process, and this issue gets worse when more than one hypothesis variable is considered.

By design, BNs are unable to explicitly encode, within their graph and statistical model, asymmetries that give rise to non-product event spaces. At the time of writing, no known extensions to the BN framework have been proposed to address this issue. We discuss this further in Chapter 4 where we explore how such asymmetries can be incorporated within a CEG model.

2.4 Chain Event Graphs

In this section, we present a non-technical description of CEGs and demonstrate, through an example, how it encodes context-specific conditional independencies and asymmetric event spaces within its graph. A formal review of CEGs is present in Chapter 3.

A chain event graph is a graphical modelling family that is represented by an acyclic directed multigraph (Smith & Anderson, 2008; Collazo et al., 2018). The construction of

a CEG for a process begins by eliciting the event tree describing the process. An event tree provides an intuitive framework for describing the evolution of a process through a sequential unfolding of events (Shafer, 1996). The transformations an event tree undergoes to become the graph of its associated CEG can be summarised as below:

- Vertices in the event tree whose one-step-ahead evolutions, i.e. conditional transition probabilities, are equivalent are assigned the same colour to indicate this symmetry;
- Vertices whose rooted subtrees (i.e. the subtree formed by considering that vertex as the root) are isomorphic – in the structure and colour preserving sense – are contracted into a single vertex which retains the colouring of its merged vertices;
- All the leaves of the tree are merged into a single vertex called the sink.

Each root-to-sink path of a CEG represents a possible trajectory of an individual within the process. We present an example below of a simplified infection process.

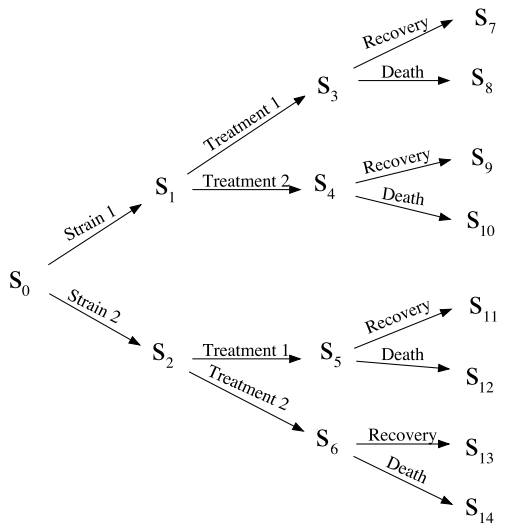


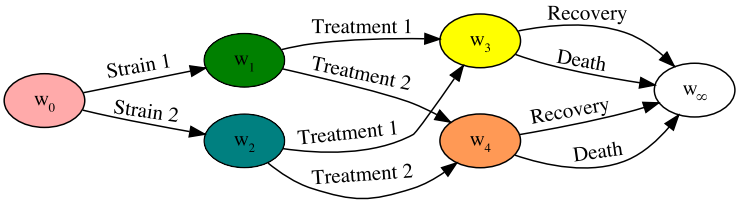
Figure 2.3: Event tree for the infection process described in Example 2.20.

Example 2.20 (Infection example)**.** *We reconsider the infection example introduced in Section 1.1. We can define three variables to describe this process:* $\boldsymbol{X} = \{X_S, X_T, X_O\}$ *where* X_S *indicates the strain of the virus with sample space* $\{Strain\ 1, Strain\ 2\}$*;* X_T *indicates the type of treatment with sample space* $\{Treatment\ 1, Treatment\ 2\}$*, and* X_O *indicates the outcome of the treatment with sample space* $\{Recovery, Death\}$*. Figure 2.3 shows the event tree for the above process. Consider the three different cases below:*

***Case 1**: The probability of recovery is independent of the strain of the virus that caused the infection, given the type of treatment administered. This can be expressed as*

$$X_O \perp\!\!\!\perp X_S \,|\, X_T.$$

This implies that the one-step-ahead evolutions of vertices s_3 and s_5, and of vertices s_4 and s_6 are equivalent. Additionally, the rooted subtrees for these pairs of vertices are isomorphic. The CEG and BN in Figure 2.4 both encode the above conditional independence relationship.



(a)

$$X_S \longrightarrow X_T \longrightarrow X_O$$

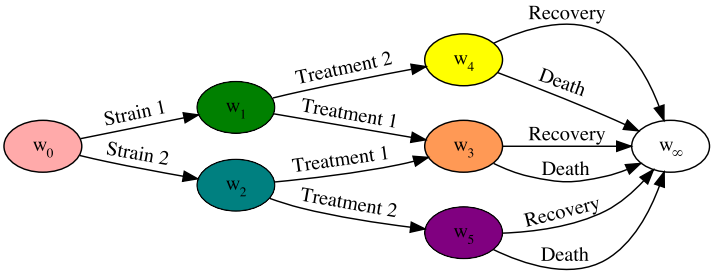
(b)

Figure 2.4: Case 1 encoded within (a) a CEG and (b) a BN.

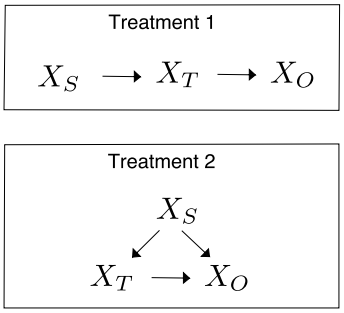
***Case 2**: The probability of recovery is independent of the strain of the virus that caused the infection given that treatment 1 has been administered but not otherwise. This can be expressed as*

$$X_O \perp\!\!\!\perp X_S \,|\, X_T = \textit{Treatment 1}$$
$$X_O \not\perp\!\!\!\perp X_S \,|\, X_T = \textit{Treatment 2}.$$

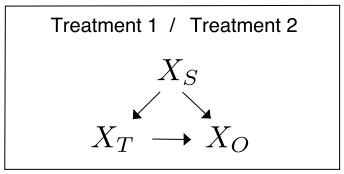
This implies that the one-step-ahead evolutions of vertices s_3 and s_5 are equivalent but s_4 and s_6 are not. This is a type of context-specific conditional independence. A standard BN cannot represent this within its graph topology. Figure 2.5 shows a CEG, a Bayesian multinet and a similarity network that can encode this context-specific relationship.



(a)



(b)



(c)

Figure 2.5: Case 2 encoded within (a) a CEG, (b) a Bayesian multinet, and (c) a similarity network.

***Case 3**: Finally, we consider here the case where treatment is only available to those who are infected with strain 1 of the virus. This could happen if the treatments were available in short supply or they were found to be ineffective against strain 2. A BN, including its variants and extensions, cannot represent this information as they all have a product event space. This case gives rise to a non-product event space as for the individuals infected with strain 2, there is no variable of treatment to be considered. Figure 2.6 shows a CEG that can represent the asymmetric event space for this process.*

2.5 Alternative Graphical Models

Finally, in this section, we briefly discuss two other modelling frameworks that could be used as alternatives to CEGs for modelling processes with asymmetric independence struc-

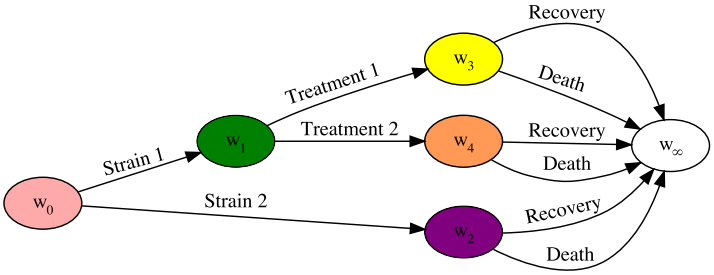


Figure 2.6: A CEG representing case 3.

tures and asymmetric event spaces.

The first of these is the *probabilistic decision graph* (PDG); introduced in Bozga and Maler (1999) for automated verification of discrete systems, and later developed as a tool for probabilistic model representation and inference in Jaeger (2004) and Jaeger et al. (2006). A PDG is, in fact, a collection of graphs – each based on an underlying tree. Although the structural syntax of these models is close to that of CEGs, there are a few significant differences. First, a CEG encodes local symmetries within its graph through its vertex colouring. Second, a CEG represents a process through a unified graph representation. Lastly, a CEG can represent context-specific conditional independencies in addition to all the conditional independencies that can be represented by a discrete BN (as discrete BNs are a special class of CEGs (Smith & Anderson, 2008)). While PDGs can represent at least some context-specific conditional independencies, there are some conditional independencies – that can be represented by BNs and CEGs – which they cannot represent within their structure (Jaeger, 2004). For instance, no PDG can represent the independence structure implied by a BN with variables $\boldsymbol{X} = \{X_1, X_2, X_3, X_4\}$ and directed edges $(X_1, X_2), (X_1, X_3), (X_2, X_4)$ and (X_3, X_4).

Finally, we discuss a family of models for discrete longitudinal data called the *acyclic probabilistic finite automata* (APFA) (Ron et al., 1998). Being developed within theoretical computer science, APFAs are formally studied as a finite state machine which is a mathematical model of computation. However, APFAs can equivalently be considered to be a PGM family. An APFA generates strings of symbols, and was developed for tasks such as speech recognition, language processing and machine translation (Edwards & Ankinakatte, 2015). Similar to CEGs, an APFA is represented by an acyclic multigraph with a single root and a single sink. Each edge of the graph is associated with a symbol

and a conditional probability. Each root-to-sink path represents a string of symbols. Like CEGs, these graphs encode context-specific conditional independencies within their topology. Ankinakatte and Edwards (2015) and Edwards and Ankinakatte (2015) showed that an APFA can be constructed from an underlying tree through vertex contractions. However, unlike CEGs, whenever two vertices are merged in the tree, their entire rooted subtrees are also merged. Vertex contractions are determined by non-Bayesian methods such as by hypothesis testing with a likelihood ratio test. Additionally, CEGs differ from APFAs as they can encode a larger set of conditional independencies, including those of the context-specific variety, through their vertex contractions and vertex colouring, and their framework allows for non-product event spaces which cannot be accommodated by the APFA framework.

Chapter 3

Chain Event Graphs

In this chapter we review the CEG family of models. CEGs were developed to overcome the drawbacks of BNs described in Section 2.3.2. Prior to the work presented in this thesis, CEG research has largely focused on the *stratified class* of CEGs for discrete processes. Hence, in this chapter, our review will be structured around stratified CEGs. The focus of the literature on the stratified class was primarily because stratified CEGs contain the class of discrete BNs as a special case (Smith & Anderson, 2008) which facilitates straightforward comparison of the two model classes. However, unlike BNs, stratified CEGs can explicitly embed context-specific conditional independencies within their graph topologies.

In Section 3.1 we review the notations and semantics of stratified CEGs. In Section 3.2, we review conjugate learning, prior specification and model selection algorithms for CEGs. Later, in Section 3.3 we review a probability propagation algorithm for CEGs. Finally, we conclude with Section 3.4 where we review dynamic variants of the CEG.

3.1 Introduction to CEGs

CEGs were first proposed in Smith and Anderson (2008) as an alternative to BNs for processes exhibiting context-specific conditional independencies with symmetric or asymmetric event spaces. Recall that contextual conditional independence exists between two sets of variables when the independence structure between them holds only for certain assignments of the conditioning set of variables. Such independencies regularly arise naturally in many applications (N. L. Zhang & Poole, 1999). As described in Section 2.3.2, BNs and their extensions are unable to fully describe context-specific independencies within their graph topologies. Below we describe how CEGs overcome these shortcomings of BNs by being a transformation of the underlying event tree of the process being modelled. Thus, their description is generally event-based rather than variable-based. However, as we shall see

later in this chapter, in the case of stratified CEGs, the event and variable based descriptions are equivalent.

In several domains such as law, forensics, risk assessment, public health interventions, and medical decision making, it is more natural for a process to be described as an unfolding of events. A domain expert in likely then to describe the process based on *how things happen*. Such a description can be most easily represented by a tree structure. In fact, it is this key point that led to the development of CEGs directly from such trees as converting a tree-like description into the alternative BN is not trivial and in several cases, results in an ill-suited BN representation of the process. A tree describing the evolution of a process is known as an *event tree*. Event trees provide an intuitive description of the evolution of a process (Shafer, 1996). This makes the elicitation of an event tree rather than the BN as the first step more natural for such processes. Note that probability trees are event trees with probabilities assigned to their edges. Decision trees, on the other hand, are tree-based decision support tools built with decision and chance nodes, and they are closely associated with influence diagrams (Howard & Matheson, 1981), see Section 2.1.

Each non-leaf vertex in the event tree represents the state an individual may be in, and its children represent the possible events that follow from this vertex. Thus, the sequence of events described by each root-to-leaf path in the event tree represents one possible way in which an individual experiences the process. Thereby the set of root-to-leaf paths of the event tree form the atoms of the event space of the process, i.e. the path $\sigma-$algebra of the event tree (Thwaites et al., 2010). We note here that as the number of events needed to describe the possible developments of a process grows, the size of its corresponding event tree (in terms of the cardinality of its vertex and edge sets) also increases. The size of the event tree increases linearly with the number of events and exponentially with the number of variables added to the description of the process. Thus, event trees may become unwieldy for large complex processes which might make them difficult to visually analyse. Nonetheless, event trees are easy for the statistician to transparently elicit from the natural language descriptions of a domain expert even if the event tree thus elicited may be very large.

Example 3.1 (Infection example)**.** *Here we build on the infection example introduced in Section 1.1. Suppose we have individuals in three residential settings: hospitals, care homes and in the general community. Further, suppose that we are interested in analysing the path an individual takes from disease onset to recovery or death on infection by one of two strains of a certain virus. The individuals receive one of two treatment types. As this problem conforms to a product space, this process can be completely described by the variables* $\boldsymbol{X} = \{X_L, X_S, X_T, X_O\}$ *where* X_L *indicates the residential setting,* X_S *the strain of virus,* X_T *the type of treatment, and* X_O *the outcome of the treatment. Note that in this*

example, there is a natural strict total order given by $X_L \prec X_S \prec X_T \prec X_O$. The event tree describing this process is given in Figure 3.1. Here, an individual at vertex s_4 is in the hospital and has strain 1 of the virus. This individual will next be given treatment 1 or 2 which is represented by the vertices s_{10} and s_{11} which are the children of vertex s_4. The sequence of events (Hospital, Strain 1, Treatment 1, Recovery) given by the path from vertex s_0 to s_{22} represents an atom of the event space of the infection process considered here.

3.1.1 Notation and Semantics

Let $\mathcal{T}$ denote an event tree with a finite vertex set $V(\mathcal{T})$ and an edge set $E(\mathcal{T})$. A directed edge $e \in E(\mathcal{T})$ from vertex v to vertex v' with edge label l is an ordered triple given by (v, v', l). Note that when it is unambiguous, more specifically the case when there is only one directed edge in a given direction between two vertices, the edge may be represented by the indices of the starting and terminating vertices, i.e. such an edge from a vertex v_i to a vertex v_j, $i \neq j$ may be denoted by e_{ij}. Observe that this is always the case for an event tree but not necessarily for its CEG which can be a multigraph.

Denote by $L(\mathcal{T})$ the set of leaves in $\mathcal{T}$. The non-leaf vertices in $\mathcal{T}$ are called *situations* and their set is denoted by $S(\mathcal{T}) = V(\mathcal{T}) \backslash L(\mathcal{T})$. The set of children of a vertex v are denoted by ch(v). Let $\Phi_{\mathcal{T}} = \{\boldsymbol{\theta}_v | v \in S(\mathcal{T})\}$ where $\boldsymbol{\theta}_v = (\theta(e) | e = (v, v', l) \in E(\mathcal{T}), v' \in \text{ch(v)})$ denotes the conditional transition parameters for each vertex $v \in S(\mathcal{T})$. A *floret* of a vertex v in $\mathcal{T}$ is denoted by $F(v) = (V(F(v)), E(F(v)))$ where $V(F(v)) = \{v \cup \text{ch}(v)\}$ and $E(F(v))$ is the set of edges induced by $V(F(v))$ in $\mathcal{T}$. Denote the set of root-to-leaf paths in an event tree $\mathcal{T}$ by $\mathcal{T}_\Lambda$ where a path is a sequence of edges from the root vertex to the leaf following the directed edges.

Example 3.2 (Infection example continued)**.** *In Figure 3.1, for situation $s_4 \in S(\mathcal{T})$ we have emanating edges $e_{4,10} = (s_4, s_{10}, Treatment\ 1)$ and $e_{4,11} = (s_4, s_{11}, Treatment\ 2)$. The floret of vertex s_4 is given by $F(s_4) = (V(F(s_4)), E(F(s_4)))$ where $V(F(s_4)) = \{s_4, s_{10}, s_{11}\}$ and $E(F(s_4)) = \{e_{4,10}, e_{4,11}\}$.*

Next we define the concept of stages which is a key concept that enables us to explore the local symmetries existing within the event tree structure. This ultimately is what allows us to have a more compact representation of the event tree in the form of the graph of a CEG.

Definition 3.3 (Stage)**.** *In an event tree $\mathcal{T}$, two situations v and v' are said to be in the same stage whenever $\boldsymbol{\theta}_v = \boldsymbol{\theta}_{v'}$. Additionally, for $\theta(e) = \theta(e')$ we require that $e = (v, \cdot, l)$ and $e' = (v', \cdot, l)$ where edge e emanates from v and e' emanates from v'.*

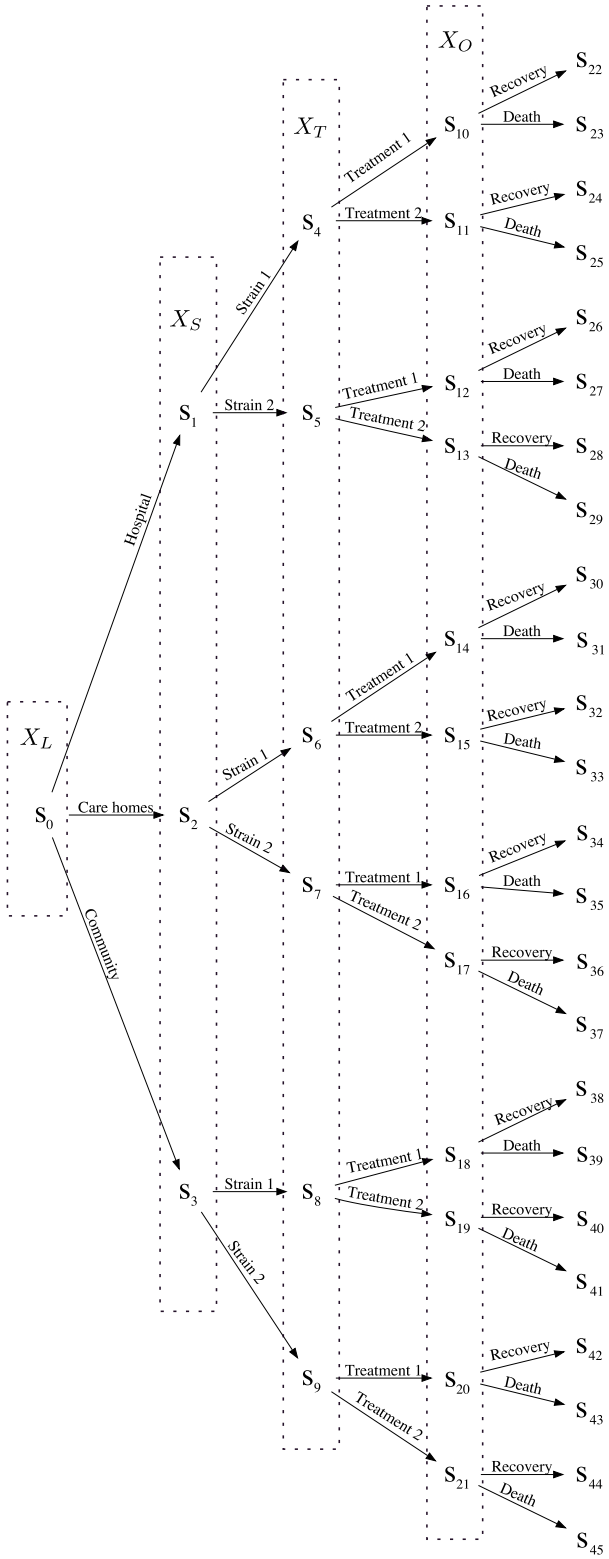
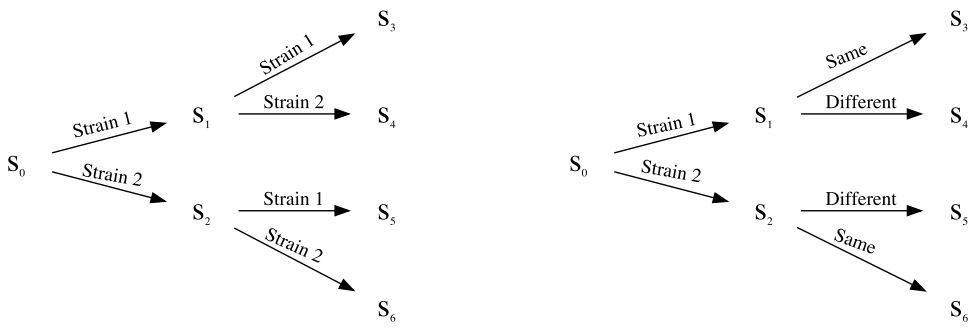


Figure 3.1: Event tree for the infection process described in Example 3.1.

The latter condition states that the edges emanating from situations in the same stage which have the same conditional transition probability must also share the same edge label. When edge labels are not fixed, this condition can be relaxed, see Shenvi and Smith (2020a). In this case, edges of vertices in the same stage are coloured to represent which edges share the same conditional transition probabilities. This allows the statistician and domain expert to retrospectively assign labels to events which have the same meaning but which could have initially been assigned different labels. For an illustration, see the example below.

Example 3.4 (Retrospective edge labelling example)**.** *Consider the event trees in Figure 3.2. Here we study the process of reinfection. If the probability of reinfection being by strain i, $i = 1, 2$, is independent of the strain of the first infection, then we could label the edges as in Figure 3.2(a). Alternatively, given the first bout of infection was by strain i, if the probability of reinfection by the same strain i is the same for $i = 1$ and $i = 2$, then it might be more appropriate to have the edge labels as in Figure 3.2(b). Note that once the appropriate edge labels are chosen, the edges lose their colouring.*



(a) (b)

Figure 3.2: Event tree where edge labels are not fixed *a priori*. (a) and (b) show two possible sets of edge labels.

As we will see in Chapter 5 in Section 5.7, edge colourings may have a different interpretation in CEGs when edges are associated with holding times. For simplicity, in this thesis we will only consider event trees where edge labels are fixed *a priori*.

Saying that two situations v and v' are in the same stage can be interpreted as asserting that their one-step evolutions are equivalent. For instance, if situations s_4 and s_5 in our infection example are in the same stage, then an individual in a hospital has the same probability of receiving Treatment 1 irrespective of whether they have Strain 1 or Strain 2 of the infection.

Observe that each stage is a set and the collection of stages partitions the vertex set of the event tree. Here a collection refers to a set of sets. Thus, the collection of stages $\mathbb{U}$ partitions $V(\mathcal{T})$ and each stage $u \in \mathbb{U}$ is a set of situations in $V(\mathcal{T})$ that belong to the stage u. Stage memberships are represented by colouring the situations of $\mathcal{T}$ such that each stage $u \in \mathbb{U}$ is represented by a unique colour.

Definition 3.5 (Staged Tree)**.** *An event tree $\mathcal{T}$ whose situations are coloured according to their stage memberships is called a staged tree $\mathcal{S}$ with $\Phi_{\mathcal{S}} = \Phi_{\mathcal{T}}$.*

Example 3.6 (Infection example continued)**.** *Suppose that recent studies lead us to deduce the following:*

$$\begin{aligned} X_O \perp\!\!\!\perp \{X_L, X_S\} \,|\, X_T &= \textit{Treatment 1} \\ \{X_T, X_O\} \perp\!\!\!\perp X_S \,|\, X_L &= \textit{Community} \\ X_S \perp\!\!\!\perp X_L \,|\, X_L &\neq \textit{Community} \end{aligned}$$

This information can be represented using the stage structure. The stage partition here is given by $\mathbb{U}$ which contains the following non-singleton sets:

$$\{s_1, s_2\}, \{s_8, s_9\}, \{s_{19}, s_{21}\}, \{s_{10}, s_{12}, s_{14}, s_{16}, s_{18}, s_{20}\}.$$

The staged tree for this example is given in Figure 3.3. Thus, conditional independence relations, including those of the context-specific nature such as the ones above, can be represented explicitly within the topology of the staged tree. Note that, to prevent visual cluttering, the colouring of trivial stages (i.e. stages which are singleton sets) is often suppressed.

Once we have the staged tree of the process, we can define the concept of positions.

Definition 3.7 (Position)**.** *In a staged tree $\mathcal{S}$, two situations v and v' are said to be in the same position whenever we have $\Phi_{\mathcal{S}_v} = \Phi_{\mathcal{S}_{v'}}$, where $\mathcal{S}_v$ and $\mathcal{S}_{v'}$ are the coloured subtrees of $\mathcal{S}$ rooted at v and v' respectively.*

This definition implies that the coloured rooted subtrees of situations which are in the same position are isomorphic in the colour-preserving and structure-preserving sense. In a non-technical sense, if situations v and v' are in the same position, their future evolutions are probabilistically equivalent. The collection of positions $\mathbb{W}$ is a finer partition of $V(\mathcal{T})$ and each position $w \in \mathbb{W}$ is a set of situations of $V(\mathcal{T})$ that belong to the position w.

Example 3.8 (Infection example continued)**.** *In the infection example, the non-trivial posi-*

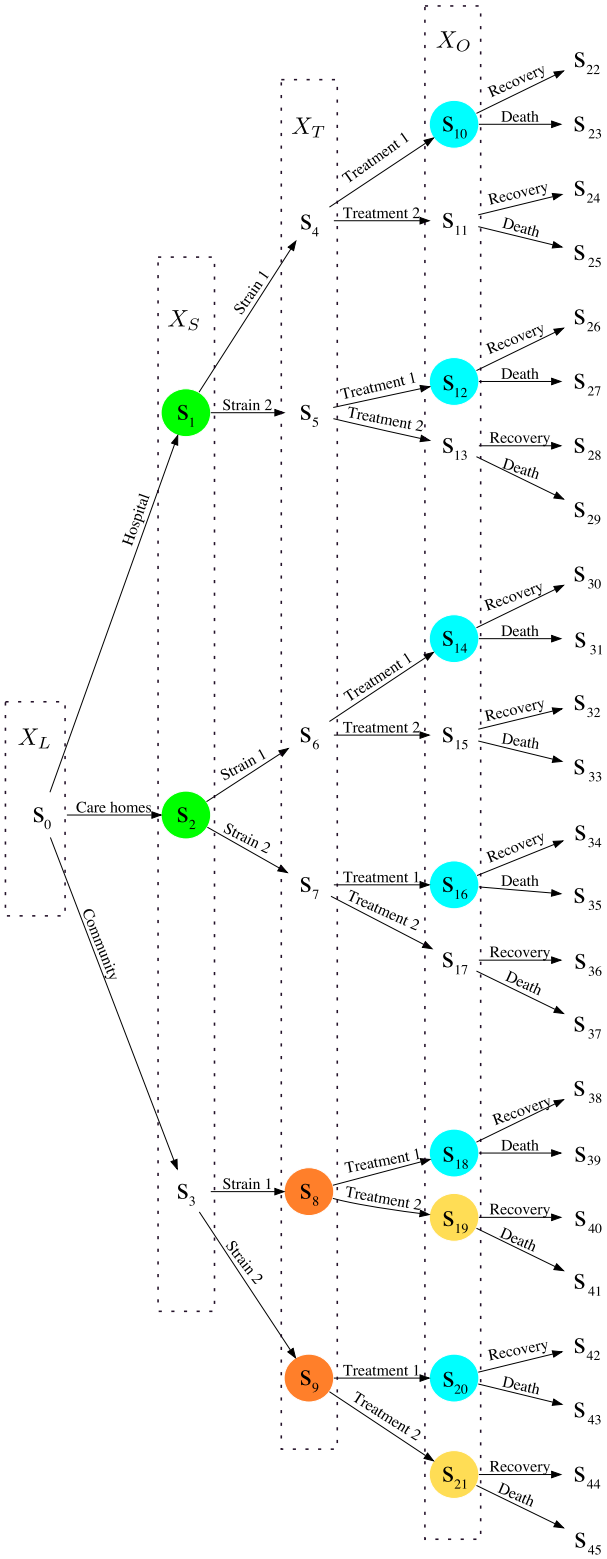


Figure 3.3: Staged tree for the infection example.

tions are given by the sets:

$$\{s_8, s_9\}, \{s_{19}, s_{21}\}, \{s_{10}, s_{12}, s_{14}, s_{16}, s_{18}, s_{20}\}.$$

Note again that stages and positions are sets. Wherever necessary, we will refer to them as a stage set or a position set to reiterate this fact.

We construct the vertex set of a CEG from its collection of positions by choosing a representative situation from each position set. Thus a CEG exploits the symmetries within the process to arrive at a compact representation of its event tree.

Definition 3.9 (Chain Event Graph)**.** *A chain event graph (CEG) $C = (V(C), E(C))$ is defined by the triple $(\mathcal{S}, \mathbb{W}, \Phi_S)$ with the following properties:*

- *$V(C) = R(\mathbb{W}) \cup w_\infty$ where $R(\mathbb{W})$ is the set of situations representing each position set in $\mathbb{W}$, w_∞ is the sink vertex and for $w \in V(C)$, $\theta_C(w) = \theta_S(w)$. Vertices in $R(\mathbb{W})$ 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 $R(\mathbb{W})$. This vertex contraction merges multiple edges between two vertices into a single edge only if they share the same edge label.*
- *Leaves of $\mathcal{S}$ are contracted into sink vertex w_∞.*

The event tree notation described earlier in this subsection extends to staged trees and CEGs in the obvious way.

Example 3.10 (Infection example continued)**.** *The graph of the CEG for the infection example is given by Figure 3.4. Here again the vertex colouring has been suppressed for vertices that represent singleton stages.*

Just as in BNs, conditional independencies, including those of the context-specific nature, can be deduced directly from the graph of the CEG without reference to the underlying parameters of the model (Smith & Anderson, 2008). There also now exists a d-separation theorem for CEGs, analogous to the one for BNs, presented in the thesis of Dr Rachel Wilkerson (Wilkerson, 2020). We note here that the CEG literature has, thus far, assumed that the transformation of a staged tree into a CEG does not lead to the loss of any information that was represented by the staged tree. This is essential to ensure that reasoning in the CEG is compatible with the information represented by its staged tree as it is typically the event tree and the staged tree that are verified to be requisite (Phillips, 1984) – that is, verified to have no obvious inadequacies in the implications of the model – by the domain experts. We show in Chapter 4 that this conjecture is true by proving that the mapping from a staged tree to a CEG is bijective. This then gives rise to the question

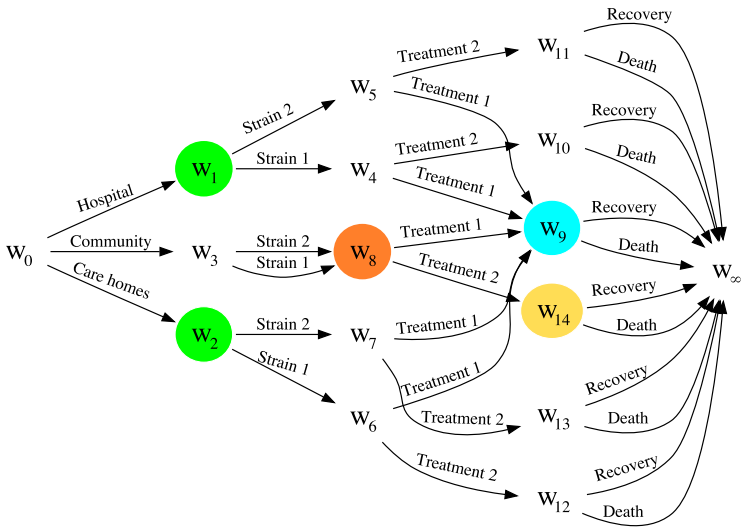


Figure 3.4: CEG for the infection example.

of why a CEG representation of a process may be useful over and above its staged tree representation since they both encode the same information. A discussion was presented in Shenvi and Smith (2020a), and is reviewed in the next subsection.

3.1.2 Why not just Staged Trees?

Before we review stratified CEGs and the existing methodologies for CEGs, we discuss here why the CEG representation is useful. Staged trees, like CEGs, are a graphical representation of a parametric statistical model and their colouring encodes conditional independence information about the events describing the process (Görgen & Smith, 2016, 2018). The thesis of Dr Christiane Görgen (Görgen, 2017) focused on the algebraic characterisation of staged trees and investigated how putative causal hypotheses can be inferred from a class of staged trees. So why do we need CEGs when staged trees are themselves powerful tools?

While we show later in Chapter 4 that the staged tree and CEG representations of a process are equivalent, the graph of a CEG is simpler and more compact. Typically, a CEG contains far fewer vertices and edges than its corresponding staged tree (Shenvi & Smith, 2020a). Let $V_k \subseteq V(\mathcal{T})$ denote the vertices of an event tree $\mathcal{T}$ with k outgoing edges and let $n_k = |V_k|$. Then $\mathcal{T}$ has $|V(\mathcal{T})| = \sum_{k=0}^{d} n_k$ vertices and $|E(\mathcal{T})| = \sum_{k=1}^{d} k n_k$ edges where $d = \max\{k : n_k \geq 0\}$. When a CEG $\mathcal{C}$ partitions V_k into $1 \leq m_k \leq n_k$ positions, it is trivial to check that it has $|V(\mathcal{C})| = \sum_{k=1}^{d} m_k + 1$ vertices (including the sink) and $|E(\mathcal{C})| = \sum_{k=1}^{d} k m_k$

edges. So we have

$$n-1 \leq |V(\mathcal{T})| - |V(C)| = \sum_{k=1}^{d} (n_k - m_k) + n - 1 \leq \sum_{k=0}^{d} (n_k - 1),$$
$$0 \leq |E(\mathcal{T})| - |E(C)| = \sum_{k=1}^{d} k\,(n_k - m_k) \leq \sum_{k=1}^{d} k\,(n_k - 1),$$

where $n = |L(\mathcal{T})|$. Let $m = \max\{|\lambda| : \lambda \in \mathcal{T}_\Lambda\}$ be the length of the longest root-to-leaf path of $\mathcal{T}$. It is easy to check that $|V(\mathcal{T})|$ and $|E(\mathcal{T})|$ typically increase as a power of m, whilst when its CEG expresses many symmetries, $|V(C)|$ and $|E(C)|$ increase linearly in m. In fact, for dynamic processes, the staged tree is infinite but the corresponding CEG might be finite as we shall see in the forthcoming chapters. Crucially, while there now exists a d-separation theorem for CEGs (Wilkerson, 2020), such methodologies are yet to be developed for staged trees.

Note that there exists an interesting framework called conditional independence trees (CITs) (H. Zhang & Su, 2004; Su & Zhang, 2005) which decompose decision trees into smaller subtrees by exploiting the conditional independence relationships (including those of the context-specific nature) exhibited by the problem. The key aim of CIT models is to provide a solution to the replication and fragmentation problems encountered by decision trees in classification. While CITs support exploration of conditional independencies, their theoretical development is still very nascent. As yet, they do not provide any formal method of causal manipulation within these models and it is hard to see how these models can be scaled to a dynamic variant. Most importantly, the representation they provide is too fragmented for a structured, unified understanding of the problem.

3.1.3 Stratified CEGs

We now describe under what conditions an event tree or CEG is said to be stratified (see e.g. Cowell and Smith (2014)). Suppose that a process can be described by a set of variables given by $\boldsymbol{X} = \{X_1, X_2, \ldots, X_n\}$. Denote by I a permutation of the indices $\{1, 2, \ldots, n\} \mapsto \{i_1, i_2, \ldots, i_n\}$. The permutation I can be used to reorder the components of $\boldsymbol{X}$ giving $\boldsymbol{X}(I) \triangleq \{X_{i_1}, X_{i_2}, \ldots, X_{i_n}\}$. Let the state space of variable X_i be given by $\mathbb{X}_i$. Let $\mathbb{X}^k(I) = \mathbb{X}_{i_1} \times \mathbb{X}_{i_2} \times \ldots \times \mathbb{X}_{i_k}$, $1 \leq k \leq n$. With this we define the concept of $\boldsymbol{X}(I)$-compatible event trees.

Definition 3.11 ($\boldsymbol{X}(I)$-Compatible Event Tree)**.** *An event tree $\mathcal{T}$ is said to be $\boldsymbol{X}(I)$-compatible for some permutation I if its vertex set $V(\mathcal{T})$ contains a root vertex s_0 as well as a vertex $s(x^k(I))$ for each $x^k(I) = (x_{i_1}, x_{i_2}, \ldots, x_{i_k})$ where $x_{i_j} \in \mathbb{X}_{i_j}$, $j = 1, 2, \ldots, k$, $1 \leq k \leq n$.*

There are two important things to notice in this definition. The first is that it implies that the event space of the process is equivalent to the product space generated by $\prod_{X_i \in \boldsymbol{X}(I)} \mathbb{X}_i$. This directly implies that in an $\boldsymbol{X}(I)$-compatible event tree the state space of

a variable remains fixed irrespective of the realisations of the other variables in the probability space. Secondly, each situation $s(x^k(I)) \in V(\mathcal{T})$ for $x^k(I) \in \mathbb{X}^k(I)$, $1 \leq k \leq n$ is at the same distance from the root s_0 where distance between vertices v and v' is measured as the number of edges on a path between them. Thus an $\boldsymbol{\mathcal{X}}(I)$-compatible event tree can be described completely by the variables in $\boldsymbol{\mathcal{X}}(I)$ where there exists a strict total order $\mathcal{X}_{i_1} \prec \mathcal{X}_{i_2} \prec \ldots \prec \mathcal{X}_{i_n}$ as determined by the permutation I.

Example 3.12 (Infection example continued)**.** *In the infection example, recall that the set of variables describing the process is given by* $\boldsymbol{\mathcal{X}} = \{X_1, X_2, X_3, X_4\}$*, where* $X_1 = X_L, X_2 = X_S, X_3 = X_T$*, and* $X_4 = X_O$*. The sample space of each variable is given as* $\mathbb{X}_1 = \{$*Hospital, Care homes, Community*$\}$ $\mathbb{X}_2 = \{$*Strain 1, Strain 2*$\}$*,* $\mathbb{X}_3 = \{$*Treatment 1, Treatment 2*$\}$ *and* $\mathbb{X}_4 = \{$*Recovery, Death*$\}$*. Recall further that the infection process naturally implies a strict total order of* $X_L \prec X_S \prec X_T \prec X_O$*. Additionally, the event space of the event tree in Figure 3.1 conforms to the product space* $\mathbb{X}_1 \times \mathbb{X}_2 \times \mathbb{X}_3 \times \mathbb{X}_4$*. Thus, the event tree in Figure 3.1 is* $\boldsymbol{\mathcal{X}}(I)$*-compatible where I is the identity permutation, i.e.* $I(i) = i$ *for* $i = 1, 2, 3, 4$*.*

In contrast, the event-based description of a process does not rely on a pre-defined set of variables. An event tree can be elicited by simply considering the unfolding of *events* along each root-to-leaf path without regard to the variables defining the process. It follows directly from the definition of an $\boldsymbol{\mathcal{X}}(I)$-compatible event tree that its variable-based and event-based descriptions are equivalent.

Definition 3.13 ($\boldsymbol{\mathcal{X}}(I)$-Stratified CEG)**.** *A CEG is said to be* $\boldsymbol{\mathcal{X}}(I)$*-stratified for some permutation I when its underlying event tree* $\mathcal{T}$ *is* $\boldsymbol{\mathcal{X}}(I)$*-compatible.*

From this definition we can see that an $\boldsymbol{\mathcal{X}}(I)$-stratified CEG also has equivalent event and variable based descriptions. Thus any $\boldsymbol{\mathcal{X}}(I)$-stratified CEG also has an associated BN on the same variables given by $\boldsymbol{\mathcal{X}}$ but not necessarily with the total order determined by $\boldsymbol{\mathcal{X}}(I)$ as the DAG of a BN determines a partial order on its variables (see e.g de Campos and Castellano (2007)). The converse is also true that any discrete state BN can be written as a stratified CEG; the proof for this result can be found in Smith and Anderson (2008). However, a BN cannot directly represent context-specific independencies within its graph topology (see Section 2.3.2) while a stratified CEG is expressive enough to do so. However, note that CEGs are their own distinct family of models and they are not just an embellishment of BNs. This fact is reinforced when we explore the modelling of processes with asymmetric event spaces with non-stratified CEGs in Chapter 4. We shall see in Section 4.2 that the event and variable based descriptions for asymmetric processes are not equivalent, and in Section 4.5 that CEGs are better-equipped than BNs for representing such processes.

3.2 Conjugate Learning and Model Selection

3.2.1 Conjugate Learning

In this section, we shall consider conjugate updating of the parameters of a CEG model. Conjugate analysis is attractive not only because it enables closed form updating of the posterior and the model marginal likelihood but also because it lends an interpretation to the hyperparameters of the priors and posteriors which can be lost when we resort to numerical methods. This review is primarily based on the work of Freeman and Smith (2011a) and Collazo et al. (2018). The methodology described here closely resembles the established framework for conjugate learning developed for discrete BNs, see e.g. Heckerman et al. (1995) and Heckerman (2008).

Consider a CEG C with collection of stages $\mathbb{U} = \{u_1, u_2, \ldots, u_k\}$. Denote by k_i the number of outgoing edges from each situation in stage set u_i, $i = 1, 2, \ldots, k$. Recall that the conditional transition probabilities from each situation within the same stage are equivalent. Let the conditional transition parameter vector for each situation in stage u_i with k_i outgoing edges be denoted by $\boldsymbol{\theta}_i = \{\theta_{i1}, \theta_{i2}, \ldots, \theta_{ik_i}\}$ where θ_{ij} is the probability that an individual in some situation $s \in u_i$ traverses along its jth outgoing edge, for $j \in \{1, 2, \ldots, k_i\}$. Assuming proper randomisation of the sampling experiment, the vector $\boldsymbol{\theta}_i$ represents the event probability parameters of a Multinomial distribution (Feller, 1971) and hence, $\sum_{j=1}^{k_i} \theta_{ij} = 1$ and $\theta_{ij} \geq 0$. Here, we will make the latter condition stricter by requiring that $\theta_{ij} > 0$, $j \in \{1, 2, \ldots, k_i\}$. Assuming that we have no sampling zeros in our data, this condition implies that there are no structural zeros in our model (see Section 4.1).

Suppose that we have a complete random sample (i.e. with no missing data) given by $\mathbf{y} = \{y_1, y_2, \ldots, y_k\}$ such that each $y_i = (y_{i1}, y_{i2}, \ldots, y_{ik_i})$ is a vector summarising the number of individuals y_{ij}, $j \in \{1, 2, \ldots, k_i\}$ that start in some situation $s \in u_i$ and traverse along its jth edge. The likelihood of the CEG C can be decomposed into the product of the likelihood of each stage floret as follows:

$$p(\mathbf{y} \mid \Phi_C, C) = \prod_{i=1}^{k} p(y_i \mid \boldsymbol{\theta}_i, C) \tag{3.1}$$

where $\Phi_C = \{\boldsymbol{\theta}_i | u_i \in \mathbb{U}\}$. Further, conditional on $\boldsymbol{\theta}_i$, the individuals in y_i are independent of each other, i.e. they are exchangeable. This makes the normalising constant equal to one for each observation in the probability mass function of the Multinomial distribution. This gives us

$$p(y_i \mid \boldsymbol{\theta}_i, C) = \prod_{j=1}^{k_i} \theta_{ij}^{y_{ij}}. \tag{3.2}$$

Analogous to the modelling assumptions of local and global parameter independence, and parameter modularity in BNs (see e.g. Spiegelhalter and Lauritzen (1990)), we assume here that the transition parameters $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_k$ are mutually independent *a priori*. Under the separability of the likelihood shown above, it follows that they will also be mutually independent *a posteriori*.

Now suppose that the CEG structure in which each situation is in a singleton stage is called C_0. Freeman and Smith (2011a) show that based on two assumptions, namely 1) *path independence:* the rates at which units traverse the root-to-sink paths in C_0 are independent, and 2) *floret independence:* the probability of units traversing an edge after reaching a situation is independent of the rate at which they arrived at the situation, that each parameter vector $\boldsymbol{\theta}_i$ associated with singleton stage u_i in C_0 has an independent Dirichlet prior. Hence, we assume here that each $\boldsymbol{\theta}_i$ has a Dirichlet prior distribution with parameter vector $\boldsymbol{\alpha}_i = (\alpha_{i1}, \alpha_{i2}, \ldots, \alpha_{ik_i})$ where $\alpha_{ij} > 0$, $j \in \{1, 2, \ldots, k_i\}$. Thus, the prior distribution of Φ_C has the following density

$$\begin{aligned} p(\Phi_C \,|\, C) &= \prod_{i=1}^{k} p(\boldsymbol{\theta}_i \,|\, C) \\ &= \prod_{i=1}^{k} \frac{\Gamma(\bar{\boldsymbol{\alpha}}_i)}{\prod_{j=1}^{k_i} \Gamma(\alpha_{ij})} \prod_{j=1}^{k_i} \theta_{ij}^{\alpha_{ij}-1}, \end{aligned} \tag{3.3}$$

where $\bar{\mathbf{v}} = \sum_{i=1}^{n} v_i$ for any vector $\mathbf{v} = (v_1, v_2, \ldots, v_n)$, and $\Gamma(z) = \int_0^\infty x^{z-1} \exp(-x)\, dx$ is the Gamma function. Thus we have

$$\begin{aligned} p(\boldsymbol{\theta}_i \,|\, y_i, C) &\propto p(\boldsymbol{\theta}_i \,|\, C) \prod_{j=1}^{k_i} p(y_{ij} \,|\, \boldsymbol{\theta}_i, C) \\ &\propto \prod_{j=1}^{k_i} \theta_{ij}^{\alpha_{ij}-1} \theta_{ij}^{y_{ij}} \\ &= \prod_{j=1}^{k_i} \theta_{ij}^{\alpha_{ij}+y_{ij}-1}. \end{aligned} \tag{3.4}$$

From this, it directly follows that $\boldsymbol{\theta}_i | y_i$ also has a Dirichlet distribution with parameter vector $\boldsymbol{\alpha}_i^* = (\alpha_{i1}^*, \alpha_{i2}^*, \ldots, \alpha_{ik_i}^*)$ where $\alpha_{ij}^* = \alpha_{ij} + y_{ij}$, $j \in \{1, 2, \ldots, k_i\}$, $i \in \{1, 2, \ldots, k\}$. Thus, under a conjugate analysis, the parameters for each stage can be updated independently in closed form.

Another implication of the conjugate analysis is that the marginal likelihood is also

available in closed form as follows

$$
\begin{aligned}
p(\mathbf{y}\,|\,C) &= \int_{\Phi_C} \prod_{i=1}^{k} \left\{ p(y_i\,|\,\boldsymbol{\theta}_i, C)\, p(\boldsymbol{\theta}_i\,|\,C) \right\} d\Phi_C \\
&= \int_{\Phi_C} \prod_{i=1}^{k} \left\{ \prod_{j=1}^{k_i} \theta_{ij}^{y_{ij}} \times \frac{\Gamma(\bar{\boldsymbol{\alpha}}_i)}{\prod_{j=1}^{k_i} \Gamma(\alpha_{ij})} \prod_{j=1}^{k_i} \theta_{ij}^{\alpha_{ij}-1} \right\} d\Phi_C \\
&= \prod_{i=1}^{k} \left\{ \frac{\Gamma(\bar{\boldsymbol{\alpha}}_i)}{\Gamma(\bar{\boldsymbol{\alpha}}_i^*)} \prod_{j=1}^{k_i} \frac{\Gamma(\alpha_{ij}^*)}{\Gamma(\alpha_{ij})} \right\}. \qquad (3.5)
\end{aligned}
$$

Here we note that model selection algorithms in BNs can be typically grouped into three broad categories: constraint-based algorithms, score-based algorithms and hybrid algorithms (Scutari, 2010, 2018). Model selection algorithms explored thus far for CEGs have been score-based (see Section 3.2.3). The general approach involves combining prior knowledge (through informative specification of the prior distributions) and data to identify the *maximum a posteriori* (MAP) network structure that has the highest score. Under a Bayesian framework for discrete state space CEGs and BNs, the MAP model is obtained by maximising one of several possible Bayesian Dirichlet (BD) scores (Heckerman et al., 1995). Other scoring functions such as the Bayesian information criterion (Schwarz, 1978), Akaike's information criterion (Akaike, 1974), factorised normalised maximum likelihood (Silander et al., 2010) could alternatively be used.

We now describe the general form of the BD scoring function which is obtained as the joint probability $p(C, \mathbf{y})$ of a CEG structure C and the observed data $\mathbf{y}$. This joint probability can be written as

$$
p(C, \mathbf{y}) = p(C)\; p(\mathbf{y}|C). \qquad (3.6)
$$

where $p(C)$ is the prior probability of the CEG structure given by model C. Equation 3.6 can also be written as

$$
\log p(C, \mathbf{y}) = \log p(C) + \log p(\mathbf{y}|C). \qquad (3.7)
$$

It is standard to assume that all CEG structures are equally likely *a priori*. In this case, we can score and compare candidate CEG models only by considering the log of the marginal likelihood $p(\mathbf{y}|C)$ given in Equation 3.5. Thus maximising the BD score is equivalent to maximising the log marginal likelihood. Henceforth, we shall refer to this as the log marginal likelihood score denoted by $Q(C)$ for a CEG C. The log marginal likelihood score

$Q(C)$ is given as follows

$$Q(C) = \sum_{i=1}^{k} \left\{ g(\bar{\boldsymbol{\alpha}}_i) - g(\bar{\boldsymbol{\alpha}}_i^*) + \sum_{j=1}^{k_i} \{ g(\alpha_{ij}^*) - g(\alpha_{ij}) \} \right\} \tag{3.8}$$

where $g(\cdot) = \log \Gamma(\cdot)$. Within the BN setting, different choices for the prior Dirichlet hyperparameters produces different priors such as the K2 score (Cooper & Herskovits, 1991), the Bayesian Dirichlet equivalent uniform score (Heckerman et al., 1995), and the more recent Bayesian Dirichlet sparse score (Scutari, 2018). Each type of prior has a different property, for example the BDeu score takes the same value for all DAGs in the same equivalence class which lends them well to causal interpretation (Heckerman et al., 1995; Pearl, 2009). In this thesis we use the general BD score function and discuss prior specification in Section 3.2.2. When we compare models using this score, the highest scoring model thus obtained is the MAP model structure.

To compare two candidate CEG models, we use the logarithm of the Bayes Factor which is simply the likelihood ratio of the log marginal likelihoods of the competing models. Hence, the log Bayes Factor of two distinct CEG models C and C' is written as

$$\log BF(C, C') = Q(C) - Q(C'). \tag{3.9}$$

Kass and Raftery (1995) provide a useful, albeit somewhat arbitrary, interpretation of the Bayes Factor as given in Table 3.1.

$\log BF(C, C')$	$BF(C, C')$	Evidence against 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: Interpretation of the Bayes Factor.

We shall now consider how this simplifies in the case of one-nested CEGs. Without loss of generality, two CEGs C and C' are said to be *one-nested* when two stages $u_i, u_j \in \mathbb{U}_C$ in C are represented by a single stage $u_{i\oplus j} \in \mathbb{U}_{C'}$ in C'. Two stages can be combined in this way if and only if their constituent situations meet the conditions under the definition of stages. Thus, the situations in u_i, u_j and $u_{i\oplus j}$ have the same number of outgoing edges denoted here by k. The log Bayes Factor will then simply be a linear combination of the

terms involving the hyperparameters associated with stages u_i, u_j and $u_{i\oplus j}$ only as given by

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

This simplifies the log Bayes Factor calculation as only terms associated with stages that are different between two models are needed.

3.2.2 Prior Specification

Setting the hyperparameters of the Dirichlet priors over all the stage parameters is a non-trivial and formidable task. There have been several choices proposed for setting the hyper-parameters of the Dirichlet priors within a BN, see for example Heckerman et al. (1995), Neapolitan (2003), and Scutari (2018). In this section we discuss two ways of specifying the hyperparameters for the Dirichlet priors within a CEG. Recall first that since the conditional transition parameters are assumed to be mutually independent, we can set the transition parameter $\boldsymbol{\theta}_i$ for stage $u_i \in \mathbb{U}$ independently of the transition parameters for the other stages in $\mathbb{U}$.

Consider, for a CEG C, the floret $F(s)$ for a situation s representing stage $u_i \in \mathbb{U}$ in the graph. Note that there can be multiple situations representing any given stage as situations constituting the same stage set are not necessarily in the same position set. However, the conditional transition parameters associated with each situation representing the stage in the CEG would be equivalent. Suppose $|E(F(s))| = k_i$, i.e. situations in stage u_i have k_i outgoing edges. Let $Dir(\boldsymbol{\alpha}_i)$ where $\boldsymbol{\alpha}_i = (\alpha_{i1}, \alpha_{i2}, \ldots, \alpha_{ik_i})$ be the prior for the parameter $\boldsymbol{\theta}_i$ associated with stage u_i. The mean vector of the Dirichlet prior is given by $\boldsymbol{\mu}_i = (\mu_{i1}, \mu_{i2}, \ldots, \mu_{ik_i})$ where $\mu_{ij} = \frac{\alpha_{ij}}{\bar{\boldsymbol{\alpha}}_i}$, $j = 1, 2, \ldots, k_i$. Thus the vector $\boldsymbol{\alpha}_i$ is completely specified by the prior means $\boldsymbol{\mu}_i$ and $\bar{\boldsymbol{\alpha}}_i$. The prior mean μ_{ij} can be interpreted as the fraction of the population expected to traverse along the jth edge emanating from s in the graph of the CEG C. This can be elicited from domain experts. We can interpret $\bar{\boldsymbol{\alpha}}_i$ are the *imaginary or equivalent sample size* (Scutari, 2018). This acts as a measure of the strength of the beliefs of the domain experts with larger effective sample sizes being associated with smaller variances on the domain experts' beliefs. Here we are effectively treating the α_{ij}'s as "pseudo-counts". However, it might not be easy or feasible to elicit the required prior mean vectors or the imaginary sample size for each stage especially when there may be several different experts contributing to the different parts of the CEG structure.

The second approach is similar to the first but requires considering the entire graph of the CEG and not just individual florets. Here we use the *mass conservation property*

(Collazo et al., 2018) which states that the number of imaginary units passing along the edges of a situation s representing the stage u_i must be equivalent to the number of imaginary units that arrive at situation s. This imaginary sample size and the way it is spread across the edges of the graph would again need to be elicited from domain experts. The difference between this approach and the previous one is that here the imaginary sample is spread across the entire graph using the mass conservation property whereas in the previous approach, each floret respects the mass conservation property but the imaginary sample size for each vertex in the graph can be chosen independently. While this approach requires us to only choose one imaginary sample size starting at the root, we still need to elicit how this sample size would be spread across the entire graph.

Using the second approach we now define a default way of setting the hyperparameters which is useful in cases where enough information or expert opinion may not be available to set appropriate hyperparameters. To do this, an imaginary sample size for the root vertex needs to be chosen. This imaginary sample size is then propagated uniformly across the edges of the CEG graph. By propagating the imaginary sample size in this way, the stages closer to the root vertex will have larger Dirichlet prior hyperparameters and smaller variances than those that are further away from the root vertex. To ensure that the default prior setting does not have a large influence on the model, we choose weakly informative priors by choosing the imaginary sample size starting at the root to be small – typically chosen as the maximum number of outgoing edges from any vertex in the graph (see e.g. Barclay et al. (2015) and Collazo et al. (2018)). However, in this default setting, vertices representing the same stage in the CEG might not have the same prior setting as shown in the example below.

Example 3.14 (Default hyperparameter setting example)**.** *Consider the CEG graph given in Figure 3.5. Here we have two vertices coloured in orange which represent the same stage but are not in the same position. We can see here how using a default prior setting gives them different Dirichlet priors with parameters* $(0.5, 0.5)$ *and* $(1, 1)$.

Two vertices representing the same stage having different priors within the CEG may be problematic as conjugate parameter updating and calculation of the model marginal likelihood (described in Section 3.2.1), are performed over each stage. Hence, care must be taken when specifying priors over stage parameters of a CEG under a default setting.

The approaches described above can also be used to set priors over the situations of the event tree. In the case of model selection, we set priors assuming that each situation in the event tree is a singleton stage. Model selection algorithms can then be used to identify the non-trivial stages as described in the following subsection.

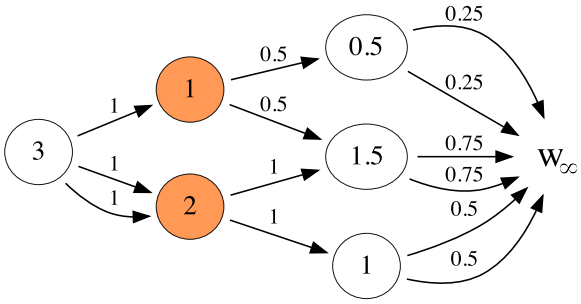


Figure 3.5: Default hyperparameter setting: the numbers represent the units of the imaginary sample arriving at the vertices (inside the circles) and passing along the edges (above the directed edges).

3.2.3 Model Selection

CEG model selection algorithms take as input the event tree of the process and output the staged tree for the process. A CEG is completely specified by its staged tree and the parameters over the staged tree $\Phi_{\mathcal{S}}$ (see Theorem 4.9 in Chapter 4). CEG model selection algorithms devised thus far are score-based and they fall under the two approaches of agglomerative hierarchical clustering (AHC) (Freeman & Smith, 2011a) and dynamic programming (Silander & Leong, 2013; Cowell & Smith, 2014). Under both these approaches, we aim to maximise the log marginal likelihood score $Q(C)$ of the CEG structure as given by Equation 3.8. The staged tree for a given process can be obtained by identifying the collection of stages that partition the situations of its event tree. Thus, finding the highest scoring CEG structure is equivalent to finding the partition of its underlying event tree's non-leaf vertices into clusters (where each cluster is equivalent to a stage) that maximises its log marginal likelihood $Q(\mathcal{T})$.

The pseudo-code for the AHC algorithm as it applies to model selection in CEGs is given in Algorithm 1. The AHC algorithm uses a bottom-up hierarchical clustering methodology beginning with the coarsest clustering treating each situation as a singleton cluster and successively merging pairs of stages until the log marginal likelihood score cannot be improved further. Thus, AHC is a local greedy search algorithm which aims to maximise the overall score by finding the next move that leads to a maximum increase in the score. Clearly, this has a downside of searching only a limited area of the model search space and can get stuck in a local maxima. For instance, a stratified CEG for a certain ordering of 4 binary variables – each with the same set of edge labels – has approximately 1.38×10^9 possible stagings but the AHC evaluates only 560 of them at most.

Note that Freeman and Smith (2011a) and in fact several other papers in the CEG

literature do not require that two situations have matching edge labels to be in the same stage. Thus, the AHC algorithm would check the log Bayes Factor of every pair of stages with the same number of outgoing edges irrespective of their edge labels. This results in an extremely large model search space with cubic time complexity while not necessarily adding more value to the interpretability of the stages thus obtained. Hence, we restrict the search space here by also requiring that situations in the same stage also have the same set of edge labels (which is consistent with the definition of stages presented in Section 3.1.1).

Algorithm 1: AHC algorithm

Input : Event tree $\mathcal{T}$, data $\mathbf{y}$, root equivalent sample size $\bar{\boldsymbol{\alpha}}_0$.
Output: Collection of stages $\mathbb{U}$, log marginal likelihood score of the MAP CEG C found by the algorithm.

Initialise an array *stages* of each situation s_i in $\mathcal{T}$.
Initialise an array *data* of y_i for each situation s_i in $\mathcal{T}$ obtained from $\mathbf{y}$.
Initialise an array *priors* of $\boldsymbol{\alpha}_i$ for each situation s_i in $\mathcal{T}$ obtained by the mass conservation property from $\bar{\boldsymbol{\alpha}}_0$.
Set *score* as the log marginal likelihood score given in Equation 3.8.
Set *indicator* $\leftarrow 1$.
while *indicator* $\neq 0$ **do**
    for *every pair of stages in stages with same number of outgoing edges and equivalent set of edge labels* **do**
        Calculate the log BF as given in Equation 3.10 comparing the structures which merge them together into one stage and keep them apart respectively, all other stages being equal.
        if *no such pair exists* **then**
            indicator $\leftarrow 0$
    for *pair* u_i *and* u_j *with the largest* $\log BF$ *score* **do**
        if $\log BF(u_i, u_j) > 0$ **then**
            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 $\boldsymbol{\alpha}_j$.
        else
            indicator $\leftarrow 0$
return *stages, score*

For the stratified class, assuming that no two variables have the same set of edge labels, running AHC on the entire event tree simplifies to running AHC on each layer of the event tree independently where a layer k is defined as all the vertices at distance k from the root vertex. In the stratified class, vertices in the same layer are associated with the same variable. This approach of decomposing a larger problem (identifying stages in the entire event tree) into smaller problems (identifying stages within a given layer) is known as

dynamic programming. Silander and Leong (2013) and Cowell and Smith (2014) describe how the dynamic programming approach can be applied to obtain the globally optimal CEG structure from an $\boldsymbol{\mathcal{X}}(I)$-compatible event tree. The pesudo-code for this is provided in Algorithm 2. The number of partitions to be evaluated for a layer with k vertices is given by the kth Bell number (Cowell & Smith, 2014). Finding the best partition quickly becomes infeasible as the event tree considered grows larger. However, the dynamic programming approach itself could be further explored with a different heuristic-based partitioning scheme for scaling to larger problems. Unlike the AHC algorithm, the dynamic programming approach does not need a fixed strict total ordering of the variables and the best variable ordering can be found as described in Silander et al. (2010) and Cowell and Smith (2014). All the examples considered in this thesis have an *a priori* determined strict ordering of the variables or events for the process.

Algorithm 2: Dynamic programming

Input : $\boldsymbol{\mathcal{X}}(I)$-compatible event tree $\mathcal{T}$ for some permutation I, data $\mathbf{y}$, root equivalent sample size $\bar{\boldsymbol{\alpha}}_0$.

Output: Collection of stages $\mathbb{U}$, log marginal likelihood score of the MAP CEG C found by the algorithm.

1 Initialise an array *stages* of each situation s_i in $\mathcal{T}$.
2 Initialise an array *data* of y_i for each situation s_i in $\mathcal{T}$ obtained from $\mathbf{y}$.
3 Initialise an array *priors* of $\boldsymbol{\alpha}_i$ for each situation s_i in $\mathcal{T}$ obtained by the mass conservation property from $\bar{\boldsymbol{\alpha}}_0$.
4 Set $score \leftarrow 0$.
5 **for** *each layer in* $\mathcal{T}$ **do**
6 By calculating the score for every possible partition, find the optimal partition of the vertices in the layer that maximises the log marginal score given in Equation 3.8 of the stages in that layer.
7 Update *stages* to add the optimal partition of the vertices in the layer.
8 Set $Q(layer)$ as the log marginal score of the layer with the optimal partition.
9 $score \leftarrow score + Q(layer)$
10 **return** *stages, score*

3.3 Probability Propagation

In this section, we discuss propagation of probabilities in a CEG. Probability propagation refers to the methodology of efficiently revising the various conditional probabilities of a model given the observation of one or more events. Under a naïve approach, this could be accomplished by obtaining the full joint distribution of the model and then revising the

probabilities using Bayes' Rule conditional on the observations (see e.g. Collazo et al. (2018)). However, such an approach is prohibitive as it requires storing a large number of computations for even moderately large processes. In the case of CEGs, it is easy to do much better than this naïve approach. For instance, propagation can be performed, in a straightforward but inefficient way, on the underlying event tree of the CEG (see Section 5.2 of Collazo et al. (2018)). However, this does not scale well and leads to several redundant calculations as it does not exploit the stage structure of the process.

Propagation algorithms aim to simplify this process by directly working with the known or estimated conditional probabilities of the model for certain types of compatible observations (the meaning of "compatible" is discussed below with reference to CEGs). Propagation of probabilities in this sense is analogous to the idea of Bayesian inference. The key difference is that the propagation algorithm takes as input a given CEG with point estimates of the conditional transition probabilities. Later in the section we discuss more about these point estimates and put them within the context of a Bayesian framework. These propagation algorithms can be used to perform inference for one individual or a set of exchangeable individuals – drawn from the population to whom the CEG applies – for whom the compatible observations have been observed.

In this section we shall review the CEG probability propagation algorithm based on Thwaites et al. (2008). This algorithm along with other special cases and lazy shortcuts are discussed in great detail in the thesis of Dr Peter Thwaites (Thwaites, 2008). This probability propagation algorithm was designed to propagate information associated with the observation of a set of transitions within a CEG. It does not consider the propagation of any temporal information that might have been observed associated with these transitions (for example, how long the individual spent at vertex v before transitioning along its ith edge). In Chapter 5, we present dynamic CEGs which evolve over continuous time with conditional holding time distributions defined over its edges. In the same chapter (in Section 5.6) we present a non-trivial novel extension of the propagation algorithm reviewed here such that it can also propagate temporal observations of holding times at various vertices of a continuous time dynamic CEG.

We now set up some terminology used with reference to propagation in this thesis. Recall that an "event" refers to the reason for a transition from one vertex to another and event descriptions are provided by edge labels in a CEG. Vertices in the graph of the CEG represent the possible states an individual might experience within the process. In this way, knowledge of the edge(s) traversed by an individual in the CEG graph is informative of the vertex that individual might be in. Let evidence $\mathscr{E}$ refer to the observed set of edges or vertices traversed or occupied by an individual in the CEG. This type of evidence may be referred to as *positive* evidence where we observe that some event has occurred or some

state has been visited as compared to *negative* evidence where we observe that an event has not occurred or a state has not been visited. We shall allow negative evidence to be included in $\mathscr{E}$ by casting it as uncertain positive evidence. Recall here that *certain* evidence is where observations occur with probability one and *uncertain* evidence is where we have a non-trivial probability distribution associated with a possible set of events or states. Further, we shall assume that the probabilities associated with the elements in a given set of uncertain evidence are always equal. Note that these properties of evidence were not explicitly set out in Thwaites (2008) and Thwaites et al. (2008) but we find it beneficial to be precise about this, especially when we address incorporating temporal evidence later in Chapter 5.

Example 3.15 (Infection example continued)**.** *Suppose we observe that an individual is not from the community, this information can be included in $\mathscr{E}$ as this individual is either in vertex w_1 or w_2 in the graph of the CEG in Figure 3.4 with equal probability.*

The final property of the observations in our evidence $\mathscr{E}$ is that each is a point observation. This implies that an observation was recorded at a specific point of time rather than over an interval of time. Within a CEG where we are not yet discussing holding times for the various situations, this property is of little importance. We shall discuss this further in Chapter 5 once we have introduced the concept of holding times within the CEG framework. See, for example, Chan and Darwiche (2005), Saria et al. (2007), and Sturlaugson and Sheppard (2016) and the references therein for their treatment of different types of evidence in inference for BNs, DBNs and CTBNs.

Assume that we have a CEG $C = (V(C), E(C))$ where the conditional transition probabilities are known. Let $\mathscr{E}$ be the set of observed evidence. The observation of a vertex $v \in V(C)$ being visited reduces the possible root-to-sink paths that the individual or group of individuals might have traversed to $\Lambda(v)$ which is the set containing all the root-to-sink paths in C passing through vertex v. Similarly, the observation of an edge $(v, v', l) \in E(C)$ being traversed reduces the possible root-to-sink paths to $\Lambda(v, v', l)$ which is the set containing all the root-to-sink paths in C passing through the edge (v, v', l). Thus, with evidence $\mathscr{E}$, we can update the topology of the CEG graph to remove the paths, if any, which are rendered impossible conditioned on the elements of $\mathscr{E}$. This corresponds to the revised probability of a path being zero conditioned on $\mathscr{E}$. Denote by $\Lambda(\mathscr{E})$ the possible root-to-sink paths in C conditioned on evidence $\mathscr{E}$. Notice that $\Lambda(\mathscr{E})$ is precisely the union of paths in $\Lambda(e)$ and $\Lambda(v)$ for every edge e and vertex v in $\mathscr{E}$.

With this we can define what we shall call an $\mathscr{E}$-reduced graph of the CEG C. Note that this is analogous to the transporter CEG in Thwaites et al. (2008) although it was defined for a now disused representation of CEGs where vertices in the CEG which represented the same stage set but were not in the same position set were connected by an undirected edge instead of being assigned the same colour, as we do now.

Definition 3.16 ($\mathscr{E}$-Reduced Graph)**.** *An $\mathscr{E}$-reduced graph for a CEG C with evidence $\mathscr{E}$ is the graph of C induced by the edges in $\Lambda(\mathscr{E})$. The $\mathscr{E}$-reduced graph inherits only the graphical structure and colouring from C and not the probabilities.*

Note that the $\mathscr{E}$-reduced graph after being populated with conditional transition probabilities generally defines the graph of some CEG. As we have defined the evidence $\mathscr{E}$ to be in terms of the vertices and edges observed, the $\mathscr{E}$-reduced graph is typically sufficient for propagating this evidence. We discuss below when this might not be the case.

If the evidence $\mathscr{E}$ is allowed to contain observations of the (potential) paths traversed by an individual or a group of exchangeable individuals, then the $\mathscr{E}$-reduced graph may be inappropriate for performing propagation. The reason for this is that the set of root-to-sink paths in the $\mathscr{E}$-reduced graph here may not be equivalent to the paths in $\Lambda(\mathscr{E})$. The two sets are unequal when conditioning on evidence $\mathscr{E}$ destroys some conditional independence relations in the original CEG C. In this case, the CEG obtained by embellishing the $\mathscr{E}$-reduced graph is not representative of the process conditioned on the evidence $\mathscr{E}$.

Example 3.17 (Infection example continued)**.** *Consider the CEG of the infection process given in Figure 3.4. Suppose that we observe that a community-dwelling individual has either been infected by strain 1 and received treatment 1 or has been infected by strain 2 and received treatment 2. We can write this evidence as*

$$\begin{aligned}\mathscr{E} = (\{&((w_3, w_8,\ \textit{Strain}\ 1), (w_8, w_9,\ \textit{Treatment}\ 1)),\\ &((w_3, w_8,\ \textit{Strain}\ 2), (w_8, w_{14},\ \textit{Treatment}\ 2))\})\end{aligned}$$

where the elements within $\{\cdot\}$ indicate uncertain evidence. The $\mathscr{E}$-reduced graph of this CEG is shown in Figure 3.6. It is clear that the $\mathscr{E}$-reduced graph does not represent the process conditioned on the observed evidence $\mathscr{E}$ as it contains the following root-to-sink paths in addition to those in $\Lambda(\mathscr{E})$:

$((w_0, w_3,\ \textit{Community}), (w_3, w_8,\ \textit{Strain}\ 1), (w_8, w_{14},\ \textit{Treatment}\ 2), (w_{14}, w_\infty,\ \textit{Recovery}))$;
$((w_0, w_3,\ \textit{Community}), (w_3, w_8,\ \textit{Strain}\ 1), (w_8, w_{14},\ \textit{Treatment}\ 2), (w_{14}, w_\infty,\ \textit{Death}))$;
$((w_0, w_3,\ \textit{Community}), (w_3, w_8,\ \textit{Strain}\ 2), (w_8, w_9,\ \textit{Treatment}\ 1), (w_9, w_\infty,\ \textit{Recovery}))$;
$((w_0, w_3,\ \textit{Community}), (w_3, w_8,\ \textit{Strain}\ 2), (w_8, w_9,\ \textit{Treatment}\ 1), (w_9, w_\infty,\ \textit{Death}))$.

Further, it fails to represent the following conditional independence relationship:

$$X_T \not\perp\!\!\!\perp X_S \mid \mathscr{E}.$$

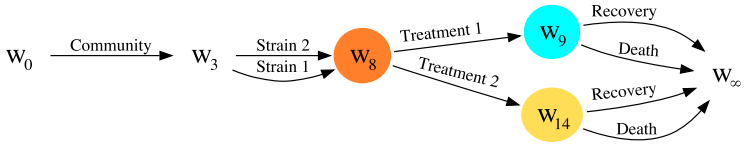


Figure 3.6: The $\mathscr{E}$-reduced graph of the CEG for the evidence described in Example 3.17.

Thus not all kinds of evidence can be propagated through the CEG. Propagation in BNs faces a similar problem when the evidence destroys some conditional independence relations in the BN. The reason for these problems is that typically propagation algorithms perform local updates that rely on the conditional independence structure of the graph. This leads us to the concept of *intrinsic evidence*.

Definition 3.18 (Intrinsic Evidence)**.** *Evidence $\mathscr{E}$ is said to be intrinsic to a CEG C if the set of root-to-sink paths in its $\mathscr{E}$-reduced graph is equivalent to $\Lambda(\mathscr{E})$.*

By defining our evidence $\mathscr{E}$ to be in terms of the vertices and edges observed, we ensure that our evidence is always intrinsic (Collazo et al., 2018).

Assume that our evidence $\mathscr{E}$ is intrinsic. Propagating the evidence $\mathscr{E}$ through the CEG C is now equivalent to populating the $\mathscr{E}$-reduced graph with conditional transition probabilities using the propagation algorithm. Thus clearly, observing some intrinsic evidence results in a graph that is, in the worst case, isomorphic to the original CEG and in most cases, smaller (in terms of number of vertices and edges) than the original CEG. In other words, observing evidence reduces the complexity of the CEG. This is the key property of propagation in a CEG that makes it more efficient compared to BN propagation.

There is another optional step of pre-processing that may be applied to the $\mathscr{E}$-reduced graph before using the CEG propagation algorithm on it. Recall that by construction, no two vertices in a CEG represent the same position set, i.e. no two vertices in the CEG have isomorphic rooted subgraphs. However, two vertices in the $\mathscr{E}$-reduced graph might have isomorphic rooted subgraphs after removal of the edges and vertices rendered impossible by the evidence $\mathscr{E}$. These vertices can then be merged to obtain the *minimal* $\mathscr{E}$-reduced graph. This does not affect the set of root-to-sink paths in the $\mathscr{E}$-reduced graph.

While the $\mathscr{E}$-reduced graph does not need to be minimal for the purposes of the propagation algorithm, we could argue that obtaining the minimal representation of the $\mathscr{E}$-reduced graph – either before or after performing the propagation algorithm – results in a graph that is a CEG as per definition. Without the minimal representation, the $\mathscr{E}$-reduced graph will be coloured to represent the stage sets and will be populated with conditional transition probabilities but each of its vertices will not represent a unique position set.

The pseudo-code for the two-pass backward-forward message-passing propagation algorithm for a CEG C and intrinsic evidence $\mathscr{E}$ as described in Thwaites et al. (2008) is presented in Algorithm 3. This algorithm has two main steps: a backward step to calculate the *potentials* and *emphases*, and a forward step which normalises the potentials to obtain the updated conditional transition probabilities. Denote the probability of occupying a vertex $v \in V(C)$ by $p(v) = p(\Lambda(v))$ and the probability of traversing an edge (v, v', l) by $p(v, v', l) = p(\Lambda(v, v', l) \,|\, \Lambda(v))$. Let $V^{-1}(v)$ denote the vertices whose emanating edges terminate in v, and $E^{-1}(v)$ denote the edges terminating in v in the CEG C. Note that $p(.)$ refers to probabilities in C and $\hat{p}(.)$ to the updated probabilities in its $\mathscr{E}$-reduced graph.

Algorithm 3: CEG propagation algorithm

Input : Conditional transition probabilities and the minimal $\mathscr{E}$-reduced graph for a CEG C and intrinsic evidence $\mathscr{E}$.
Output: Updated conditional transition probabilities.

Set $A \leftarrow \emptyset$, $B \leftarrow \{w_\infty\}$, $\Phi(w_\infty) \leftarrow 1$.
while $B \neq \{v_0\}$ *(the root vertex)* **do**
  for $v_j \in B$ **do**
    for $v_i \in V^{-1}(v_j)$ **do**
      for $e \in E(v_i) \cap E^{-1}(v_j)$ **do**
        if $e \in \Lambda(\mathscr{E})$ **then**
          $\tau_e \leftarrow p(e) \cdot \Phi(v_j)$
        else
          $\tau_e \leftarrow 0$
        $A \leftarrow A \cup \{e\}$
      if $E(v_i) \subset A$ **then**
        $\Phi(v_i) = \sum_{e \in E(v_i)} \tau_e$
        $B \leftarrow B \cup \{v_i\}$
    $B \leftarrow B \backslash \{v_j\}$
for $v_i \in V(C)$ **do**
  for $e \in E(v_i) \cap \Lambda(\mathscr{E})$ **do**
    $\hat{p}(e) = \dfrac{\tau_e}{\Phi(v_i)}$
  for $e \in E(v_i) \backslash \Lambda(\mathscr{E})$ **do**
    $\hat{p}(e) = 0$
return *Updated conditional transition probabilities* $\hat{p}(.)$

By populating the edges in the minimal $\mathscr{E}$-reduced graph with the updated conditional transition probabilities, we obtain the $\mathscr{E}$-reduced CEG which results from propagating intrinsic evidence $\mathscr{E}$ through the original CEG C. In Chapter 5 we revisit this algorithm and extend it to propagate temporal evidence.

Finally, we discuss probability propagation in CEGs when the conditional transition probabilities are estimated. Within the setting where these probabilities are estimated, we can substitute the known probabilities in the propagation algorithm above with the posterior means of the estimated conditional transition probabilities. Consider a vertex w in a CEG C with estimated posterior distribution for parameter vector $\boldsymbol{\theta}$ given by $Dir(\boldsymbol{\alpha}^*)$. Suppose that we observe an intrinsic event A that results in certain edges emanating from vertex w having a probability zero of being traversed. Conditioned on event A, the updated parameter subvector $\hat{\boldsymbol{\theta}}_A \subset \boldsymbol{\theta}$ for vertex w has a $Dir(\boldsymbol{\alpha}^*_A)$ distribution where $\boldsymbol{\alpha}^*_A \subset \boldsymbol{\alpha}^*$ (Collazo et al., 2018). Further, observe that the posterior mean vector of $\hat{\boldsymbol{\theta}}_A$ can be obtained as follows

$$\begin{aligned}\mathbb{E}[\hat{\boldsymbol{\theta}}_A] &= \frac{\boldsymbol{\alpha}^*_A}{\sum \boldsymbol{\alpha}^*_A} \\ &= \frac{\boldsymbol{\alpha}^*_A}{\sum \boldsymbol{\alpha}^*} \frac{\sum \boldsymbol{\alpha}^*}{\sum \boldsymbol{\alpha}^*_A} \\ &= \frac{\mathbb{E}[\boldsymbol{\theta}_A]}{\sum \mathbb{E}[\boldsymbol{\theta}_A]}. \end{aligned} \tag{3.11}$$

In a similar way, provided that the individual or group of individuals for whom the evidence has been observed have been drawn randomly from our population, the revised probabilities obtained from the propagation algorithm using the posterior conditional means are the expectations of the updated posterior conditional transition distributions for these individuals (Collazo et al., 2018).

3.4 Dynamic Variants of CEGs

A CEG, as described thus far, has an underlying event tree which is finite. Recall that a CEG is obtained through a colouring and transformation of its underlying event tree. An event tree supposes a strict total ordering of the events along each of its root-to-leaf paths. The CEG inherits this strict total ordering of the events along each of its root-to-sink paths. Hence, a CEG – even if it has an underlying finite event tree – cannot typically be considered to be "static". This is in contrast to BNs where BNs provide static representations of the state of a system at a fixed point in time, and its dynamic variants such as DBNs and CTBNs represent the longitudinal evolution of a system though discrete and continuous time respectively.

We shall say that a CEG is *dynamic* when its underlying event tree is infinitely large, we otherwise call it a CEG or a *vanilla* CEG (so as to not confuse with the CEG family whenever it is ambiguous). Thus, the atoms of the event space defined by the event tree of a dynamic variant of a CEG are infinite. In this section, we review the two main dynamic variants of CEGs: discrete time dynamic CEGs (DCEGs) and extended DCEGs as presented in Barclay et al. (2015). In particular, these dynamic variants of CEGs allow for individuals to experience an event more than once. Note that Freeman and Smith (2011b) developed a dynamic extension of CEGs which do not have underlying infinite event trees. This dynamic extension has a fixed underlying finite event tree but the stage set of the event tree and hence, its staged tree and CEG are allowed to change across each discrete time step. It analyses how symmetries within a process change over time.

The discrete time DCEG (or simply, the DCEG) as presented in Barclay et al. (2015) is the simplest dynamic extension of the vanilla CEG. It models a discrete state space longitudinal process. Similar to the vanilla CEG, it does not involve any explicit modelling of holding times at its various states (vertices). The implicit assumption in DCEGs is that the holding times at its various states are governed by a geometric distribution (a detailed discussion is presented in Section 5.1). This can be seen by the relationship between DCEGs and Markov chains described in Barclay et al. (2015). Stages and positions are defined for DCEGs exactly as they are for CEGs.

Definition 3.19 (Dynamic Chain Event Graph)**.** *A discrete time dynamic chain event graph (DCEG)* $\mathcal{D} = (V(\mathcal{D}), E(\mathcal{D}))$ *is defined by the triple* $(\mathcal{S}, \mathbb{W}, \Phi_{\mathcal{S}})$*, where* $\mathcal{S}$ *has an infinitely large vertex set, with the following properties:*

- $V(\mathcal{D}) = R(\mathbb{W}) \cup w_\infty$ *if* $L(\mathcal{S}) \neq \emptyset$ *and* $V(\mathcal{D}) = R(\mathbb{W})$ *otherwise, where* $R(\mathbb{W})$ *is the set of situations representing each position set in* $\mathbb{W}$*,* w_∞ *is the sink vertex and for* $w \in V(\mathcal{D})$*,* $\theta_{\mathcal{D}}(w) = \theta_{\mathcal{S}}(w)$*. Vertices in* $R(\mathbb{W})$ *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* $R(\mathbb{W})$*. This vertex contraction merges multiple edges between two vertices into a single edge only if they share the same edge label.*
- *Leaves of* $\mathcal{S}$*, if any, are contracted into sink vertex* w_∞*.*

It is not necessary that the infinite event tree underlying a DCEG has any terminating root-to-leaf walks. In fact, it may not have any leaves at all. In this case the corresponding DCEG would not have a sink vertex. Further, unlike vanilla CEGs, the graph of a DCEG may have directed cycles and it may contain loops. While the definition above does not require $V(\mathcal{D})$ to be finite, Barclay et al. (2015) considers only DCEGs whose graphs have a finite vertex set. The DCEG class is particularly useful for modelling processes where

observations are recorded at regular discrete time intervals (e.g. hourly, weekly or monthly). Barclay et al. (2015) show that DBNs are a special subclass of DCEGs. We note here that Collazo and Smith (2018a, 2018b) explored a special class of DCEGs called the N time-slice DCEG which is closely associated with the N time-slice DBN. An N time-slice model is one where the dependence structure has an Nth-order Markov property, i.e. probabilities associated with events in a time-slice $t \geq N$ are dependent only on the events that take place in the previous N time-slices.

It is of particular interest to study dynamic variants of CEGs which allow holding times to be non-geometric (in case of discrete time processes) or non-exponential (in case of continuous time processes). Not only does this enable us to model processes where observations may not be recorded at regular time intervals (for instance, observations may be recorded as events occur such as symptoms being recorded in a patient's medical history as they develop) but it also enables us to incorporate important temporal information explicitly within the model. In several domains such as medicine, reliability engineering and law, it is of interest to study *what happens next* as well as *when it happens*. The 2020/21 pandemic is a good example of such a process. Extended DCEGs are the only members of the CEG family thus far which evolve in continuous time and explicitly model holding time distributions for each event. They do so by associating a conditional holding time random variable with each edge. A holding time along edge (v, v', l) indicates how long an individual spends in the state represented by situation v before transitioning along this edge to situation v'. We now define the extended DCEG exactly as presented in Barclay et al. (2015).

Definition 3.20 (Extended Dynamic Chain Event Graph (Barclay et al., 2015))**.** *An extended DCEG $\mathcal{D} = (V(\mathcal{D}), E(\mathcal{D}))$ is a DCEG with no loops from a position into itself and with conditional holding time distributions conditioned on the current stage, u, and the next edge e_{uj}, to be passed through:*

$$F_{uj}(h) = p(H_{uj} \leq h | u, e_{uj})$$

for $h \geq 0, u \in \mathbb{U}$ and where j indexes the jth edge emanating from u, $j = 1, \ldots, m_u$. Hence, $F_{uj}(h)$ describes the time a unit stays in any position w merged into stage u before moving along the next edge e_{wj}.

The extended DCEG is a useful subclass especially when we are dealing with processes that have observations recorded at irregular times. Barclay et al. (2015) presented a semi-Markov representation of the extended DCEG for the special case where the graph of the extended DCEG is a simple graph. This enables us to use well-developed semi-Markov technologies for these models.

However, the extended DCEG subclass as presented in Barclay et al. (2015) has

some critical limitations. The focus of Barclay et al. (2015) was largely on extending the CEG with the discrete time DCEG class described earlier. The work presented there on extended DCEGs was preliminary. The extended DCEG class and methodologies for this class were not fully explored. Nor were any real-world examples explored in detail. From the definition above, we can see that the extended DCEG was presented as an extension of the DCEG class. Hence, the definition of the extended DCEG builds directly on that of a DCEG. This does not clearly describe how the concept of holding times is incorporated with stages and positions which makes it difficult to understand how a modeller should proceed to construct such an extended DCEG starting from the event tree of a process.

Further, in Barclay et al. (2015), it is assumed under a *time-homogeneity assumption* that the conditional holding time distributions for two situations are the same whenever they belong to the same stage set. This is a very prohibitive assumption as it excludes the possibility of two situations sharing equivalent conditional transition probabilities but having different holding time distributions. For example, the time it takes an individual to transition from being susceptible to infected could be dependent on the strain of the virus to which they were exposed even if their probability of being infected by either strain is the same. In Chapter 5 we present a general class of continuous time DCEGs (CT-DCEGs) that contains extended DCEGs as a special case. Another subclass of this general CT-DCEG class, known as the reduced DCEG, is developed and applied to a policing application in Chapter 6.

Chapter 4

Non-Stratified Chain Event Graphs

We reviewed stratified CEGs in Chapter 3 which contain discrete BNs as a special case. We begin this chapter by motivating the class of non-stratified CEGs in Section 4.1. Here we discuss the type of issues that result in processes having non-product event space structures and demonstrate how these can be easily accommodated within the event tree framework. In Section 4.2 we formally define a non-stratified CEG and discuss why an event-based, rather than a variable-based, formulation is more appropriate for this class. In Section 4.3 we then present a general algorithm with an optimal stopping criterion to construct a CEG from a staged tree irrespective of whether it is stratified or non-stratified. Here we also prove that no information is lost in the transformation of a staged tree into a CEG. This result has been assumed in the CEG literature so far but a formal proof had not been presented. We next describe in Section 4.4 how CEG model selection algorithms (see Section 3.2.3), developed primarily for the stratified class, can be adapted in a straightforward manner to apply to the non-stratified class. In Section 4.5 we present an application of the non-stratified CEG class to a simulated dataset of a public health intervention. We conclude with a discussion in Section 4.6.

4.1 Motivation and Introduction

To motivate the importance of the non-stratified class of CEGs, we introduce here a complex intervention designed to reduce fall-related injuries in the elderly (here on referred to as the *falls intervention*) which was presented in Eldridge et al. (2005). Note that intervention here refers to policies developed to improve physical and/or mental health of individuals within a population rather than the "intervention" of setting a variable X to a value x denoted by $do(X = x)$ within Judea Pearl's causal algebra (Pearl, 2009).

We first describe the falls intervention as presented in Eldridge et al. (2005) along

with the need for such an intervention. According to the World Health Organisation (World Health Organization, 2018): *"A fall is defined as an event which results in a person coming to rest inadvertently on the ground or floor or other lower level."* Falls-related injuries are a serious problem among the elderly with consequences ranging from fear of falling and fractures to death. Such injuries may contribute to increased morbidity, reduced mobility, possible hospitalisations and increased costs of health. According to NICE guidelines (NICE: Guidance and Guidelines, 2013), 30% of people older than 65 and 50% of people older than 80 fall at least once a year. The falls intervention was designed to enhance assessment, referral pathways and treatment for high risk individuals aged over 65 years living in the community as well as those in care homes, nursing homes and hospitals who have a substantial risk of falling. Under this intervention, a certain proportion of individuals aged 65+ would be assessed. Assessment would be carried out as per the recommendations in the Falls Risk Assessment Tool (FRAT) (Nandy et al., 2004) which classifies an individual as low or high risk based on factors such as their history of falling in the previous year, number of prescription medicines taken per day, diagnosis of stroke or Parkinson's disease etc. Those assessed to be at a high risk could then be referred to a falls clinic for an advanced assessment. It is assumed that all those who are referred, 50% of other high risk individuals, and 10% of low risk individuals would receive treatment. Further, it is assumed that those who are not assessed would receive neither referral nor treatment. The event tree for this process is shown in Figure 4.1. Note here that the choice of events to represent this process was informed by the tree-like description of the intervention presented in Figure 4 of Eldridge et al. (2005).

For our discussion we find it helpful to define variables under which each of the events occur. Consider the variable set $\boldsymbol{X} = \{X_A, X_{Ri}, X_{Re}, X_T, X_F\}$ where X_A indicates whether an individual aged 65+ is assessed or not; X_{Ri} indicates their risk level as high or low; X_{Re} indicates whether they are referred; X_T indicates whether they are treated; and X_F indicates whether they have a fall. By the design of the intervention, we have that $p(X_{Re} = \text{Not referred}|X_{Ri} = \text{Low}) = 1$; i.e. we do not observe any low risk individuals who are referred, irrespective of the sample size. Similarly, $p(X_T = \text{Treated}|X_{Re} = \text{Referred}) = 1$. Observe that $p(X_{Re} = \text{Not referred}) \neq 1$ and $p(X_T = \text{Treated}) \neq 1$. This implies that the categories of $X_{Re} = \text{Referred}$ and $X_T = \text{Not treated}$ are not redundant for all groups of individuals as can be clearly seen in Figure 4.1.

The phenomenon observed here is known as a *structural zero*. A structural zero refers to observing zero frequencies for a count variable or a category of a categorical variable when a non-zero observation is a logical impossibility (e.g. days or amount as low, medium, high of alcohol consumption by teetotallers). This is in contrast to a *sampling zero* where a zero frequency is observed due to limitations of sampling (see e.g. Mohri

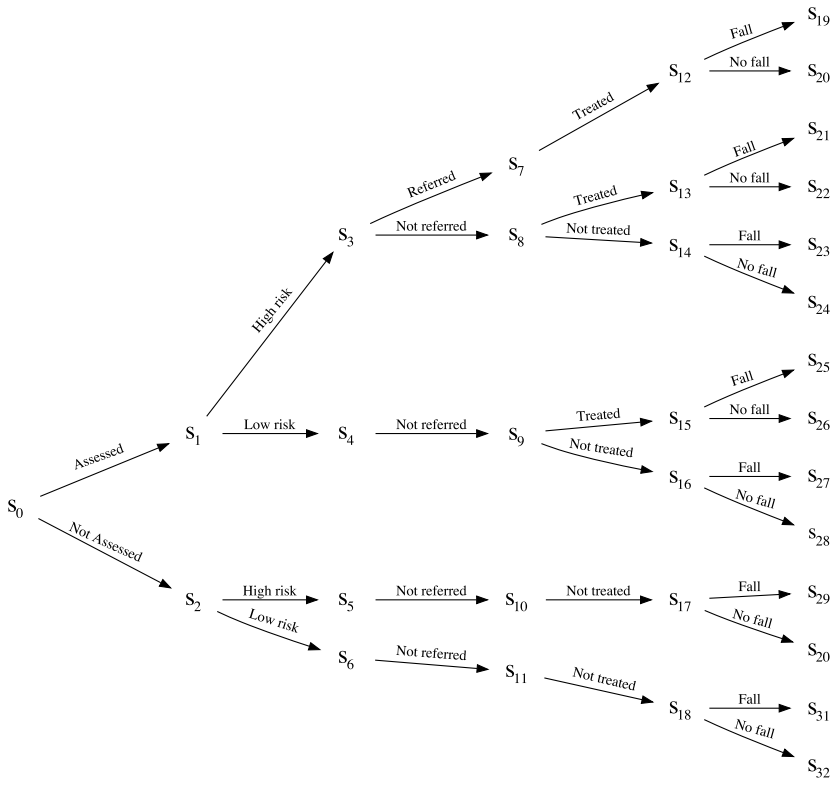


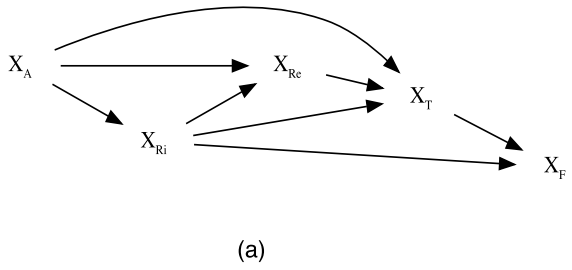
Figure 4.1: Event tree for the falls intervention.

and Roark (2005)). Inclusion of structural zeros within a model leads to storage of redundant information, and may bias the inference (Manrique-Vallier & Reiter, 2014). Within a graphical modelling framework, explicit exclusion of structural zeros within the graph of the model is also useful from the perspective of representation. Sampling zeros, on the other hand, must not be excluded from the model. Determining whether the observation of a zero frequency for an event is a structural or a sampling zero is dependent on the context of the application and generally cannot be determined solely from the data. Mohri and Roark (2005), however, describe how standard statistical criteria such as mutual information and the log odds ratio may be used to determine whether zero observations of a sequence of two or more words within a large corpora of text can be considered to be structural zeros.

Fortunately within an event tree, structural zeros can be easily accommodated by simply deleting the edges where they occur. By having an edge labelled "Treated" but not one labelled "Not treated" from situation s_7 in Figure 4.1, it is clear that those who are assessed, classified as high risk and have been referred are always treated. In this way, structural zeros are explicitly represented within the topology of the event tree. A CEG

inherits this property as it is a transformation of its underlying event tree (see Section 3.1.1).

In contrast, in a BN, these structural zeros would be hidden within its conditional probability tables (CPTs). To see this, we hypothesise the BN structure in Figure 4.2(a) for the falls intervention based on the dependencies built into the intervention by design. The table in Figure 4.2(b) gives the CPT for the variable X_{Re} which shows how these structural zeros are encoded by BNs.



(a)

X_A	X_{Ri}	p(X_{Re} = Referred)
Assessed	High	$\theta(s_3, s_7, \text{Referred})$
Assessed	Low	0
Not assessed	High	0
Not assessed	Low	0

(b)

Figure 4.2: (a) Hypothesised BN for the falls intervention; (b) CPT for variable X_{Re}.

Just as in the falls intervention example, processes within several domains such as public health, forensic science, policing and reliability theory may have structural zeros present by design. Furthermore, we observed another interesting phenomenon that occurs frequently in such domains and contributes to processes having a non-product event space. We introduce this phenomenon through the modification of the falls intervention below.

Consider a new variable X_{Rev} which indicates the revised risk of an individual after they have received treatment as high or low. Clearly, this variable is not defined for those who did not receive any treatment. Thus, for a subgroup of our population – precisely for those who receive treatment – the defining variables for this process are given by $\boldsymbol{X}' = \{X_A, X_{Ri}, X_{Re}, X_T, X_{Rev}, X_F\}$ whereas for those who do not receive treatment, these are given by $\boldsymbol{X}$. This phenomenon is known as *structural missing values*. Structural missing value are observations which are missing as they are not defined for a subset of the individuals (e.g. variables relating to post-operative health of individuals who had the illness but weren't operated). Similar to structural zeros, inclusion of structural missing values in a model leads to storage of redundant information and may bias inference. Care must be taken here to ensure that exclusion of any missing values is done after confirming that they are indeed structural. For handling of non-structural missing values in a CEG, see Barclay et al. (2014).

Encoding structural missing values within an event tree is straightforward. Florets representing structural missing values are simply excluded from the event tree. For instance,

while there would be a floret representing X_{Rev} at the end of the directed edges labelled "Treated" in Figure 4.1, there would be no analogous florets at the end of the directed edges labelled "Not treated". Just as with the structural zeros, CEGs inherit the ability to express structural missing values from their underlying event trees. In contrast, structural missing values cannot be explicitly encoded within a BN model as it assumes an event space with a product space structure (see Section 2.3.2).

We have illustrated the ease with which event trees and their corresponding CEGs can encode structural zeros and structural missing values (collectively referred hereafter as structural asymmetries). However, it is only the non-stratified class and not the stratified class that can handle these structural asymmetries. Observe that the stratified class by definition does not even accommodate event trees and CEGs where the ordering of the variables may not be the same across the tree (e.g. individuals presenting with a certain set of risk factors may be given treatment before being referred to a falls clinic for further support, whereas this order might be swapped for others who have a different set of risk factors). Further observe that the stratified class is in fact a special case of the non-stratified class.

Thus far, most of the CEG research has focused on the stratified class due to its correspondence to BNs. However, the falls example demonstrates that public health interventions may belong to the non-stratified class. Furthermore, we conjecture that structural asymmetries are the norm rather than the exception when it comes to processes that are best described through an unfolding of events as is the case in several domains such as medicine, risk analysis, policing, forensic science, law, ecology, and reliability engineering. This necessitates the development of the non-stratified class of event trees and CEGs.

4.2 Non-Stratified CEGs

As described in the previous section, event spaces that have a non-product space structure may arise due to structural asymmetries. Unlike BNs, the CEG framework can easily model such processes and can explicitly represent these asymmetries within its graph. We now present a simple definition of non-stratified CEGs – the class of CEGs that can model asymmetric processes.

Definition 4.1 (Non-Stratified CEG)**.** *A CEG C is said to be non-stratified when its underlying event tree $\mathcal{T}$ is not $\boldsymbol{\chi}(I)$-compatible where $\boldsymbol{\chi}$ is the set of variables to which the events of $\mathcal{T}$ belong and $\boldsymbol{\chi}(I)$ is any permutation of the variables in $\boldsymbol{\chi}$.*

From this definition, we can see that the class of non-stratified CEGs is extremely large. It generalises the class of stratified CEGs. In particular, this implies that the model search space of a non-stratified CEG can be very large. In Section 4.4 we introduce some

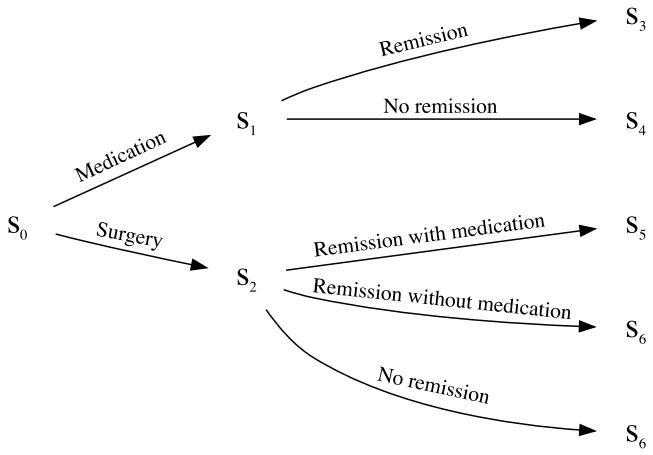


Figure 4.3: Event tree for the epilepsy example.

simplifying assumptions to restrict the model search space of non-stratified CEGs to make model selection feasible.

We next illustrate, through an example, that for non-stratified CEGs, the event-based and variable-based descriptions of the process are not equivalent.

Example 4.2 (Event-based description example)**.** *Consider a study of individuals with epilepsy who have not achieved a 12-month remission or any significant reduction of seizures after trying two anti-epileptic drugs. Further assume that these individuals suffer from focal seizures, that is seizures that begin in a specific area (lobe) of the brain. These individuals can now either opt for a treatment by another anti-epileptic drug or a surgery. After undergoing a surgery, the individuals might be able to achieve remission with or without taking another anti-epileptic drug or they might not achieve remission. Suppose that the study analyses the outcomes of the individuals based on these two options available to them. Suppose that the event tree in Figure 4.3 describes this process.*

It is clear that this process cannot be easily expressed with a variable-based description that generates an event space that admits a product space structure. To see this, we could define the variables of interest to be the following

- *a treatment variable* $X_T = \{Medication, Surgery\}$*;*
- *an outcome variable for those opting for medication* $X_{O1} = \{Remission, No\ remission\}$*;*
- *an outcome variable for those opting for surgery* $X_{O2} = \{Remission\ with\ medication, Remission\ without\ medication, No\ remission\}$*.*

Here, X_{O1} *is not defined for those who opt for surgery while* X_{O2} *is not defined for those who*

opt for another anti-epileptic drug. This results in structural missing values for variable X_{O1} among those individuals at vertex s_2, and for variable X_{O2} among those at vertex s_1. To force a variable-based description, we could combine X_{O1} and X_{O2} such that we use X_{O1} or X_{O2} for all individuals irrespective of whether they opt for another anti-epileptic drug or a surgery. However, using only X_{O1} will loss of information (by not including the distinction between remission with and remission without medication for those who opt for a surgery) and using only X_{O2} will lead to structural zeros. Hence, for such asymmetric processes, an event-based description is more appropriate.

4.3 Construction of Non-Stratified CEGs

We now consider the task of constructing the graph of a CEG C from any staged tree $\mathcal{S}$ irrespective of whether it is stratified. Observe that model selection algorithms for the CEG family take as input its underlying event tree and return its associated staged tree. That is, the output of any model selection algorithm for a CEG is a collection of stages $\mathbb{U}$ for its underlying event tree. In this section, we assume that we are only given the staged tree – obtained either as an output of a model selection algorithm or elicited by domain experts – from which we can deduce the collection of stages $\mathbb{U}$. Recall that the vertex set of a CEG C is given by $V(C) = R(\mathbb{W}) \cup w_\infty$ where $R(\mathbb{W})$ is the set of situations representing each position set in the collection of positions $\mathbb{W}$, and w_∞ is the sink vertex. Thus constructing a CEG from its staged tree entails identifying the position sets of the staged tree followed by iterative vertex contractions as defined in Section 3.1 to reduce $V(\mathcal{S})$ to $V(C)$.

4.3.1 Motivating the Construction Algorithm

The algorithm to construct a CEG C from a staged tree $\mathcal{S}$ is based on three key observations. We first motivate these key observations and the steps of the construction algorithm with an example.

Example 4.3 (Infection testing example)**.** *Consider the staged tree in Figure 4.4 which shows a hypothesised example of testing for a certain disease available to individuals exhibiting symptoms in two different settings: in hospitals and in the general community. For simplicity, we assume here that the test is 100% sensitive and specific, and that we are only interested in the outcomes related to the disease. We further assume that death can only be caused by the disease in the time period considered.*

From the staged tree for this process and through a simple visual analysis of the

graph topology, we can see that its non-trivial stage and position sets are as follows

$$\{s_3, s_{11}\},\ \{s_4, s_5, s_{13}\}.$$

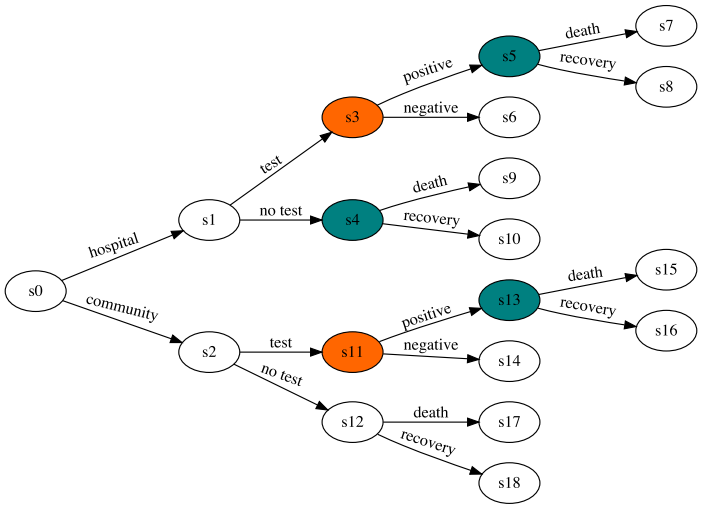


Figure 4.4: Staged tree for the infection testing example.

Key observation 1: Situations in the same position have isomorphic subtrees. Hence, they are necessarily at the same distance from a leaf of the staged tree. Here distance is measured in terms of the maximum number of edges along any directed path from the situation to any leaf vertex. This implies that a CEG can be obtained from its staged tree through a backward iteration over its situations.

Example 4.4 (Infection testing example continued)**.** *As the first step of the backward iteration, we combine all the leaves of the staged tree of the infection testing process into a single sink vertex as shown in Figure 4.5.*

For the second step of the backward iteration we merge together the situations that are at a maximum distance of one edge from the sink vertex and are in the same stage as this implies that are also necessarily in the same position. For step k of the backward iteration, we wish to identify the situations that are in the same position and are at a maximum distance of $k-1$ edges from the sink vertex, for $2 \leq k \leq m$ where m is the length of the longest root-to-leaf path in the staged tree of the process. We observe the next two steps of the backward iteration for the testing example before stating our second key observation.

Example 4.5 (Infection testing example continued)**.** *At the second step of the backward iteration, we consider the situations s_4, s_5, s_{12} and s_{13} as they are all at the maximum of*

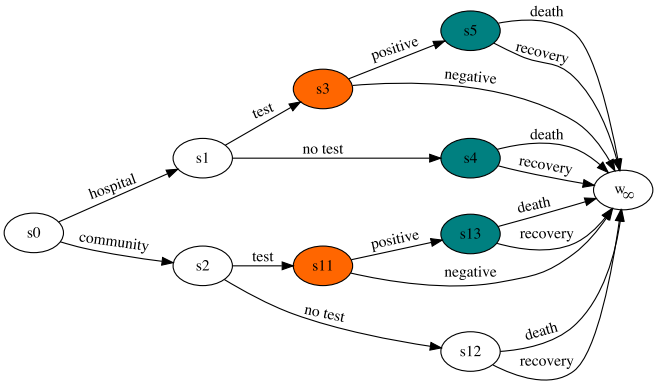


Figure 4.5: First step of the backward iteration.

one directed edge away from the sink vertex. Of these, situations s_4, s_5 and s_{13} belong to the same stage and so they are contracted into a single vertex as shown in Figure 4.6. At the third step, we consider situations s_3 and s_{11}. From the staged tree of this process, it is clear that s_3 and s_{11} are in the same position and so they are contracted into a single vertex in the third backward step. Observe that as the two situations are in the same position, they are also in the same stage. Further, observe that their outgoing edges in Figure 4.6 which share the same edge label terminate in the same vertex. Here, the edges labelled "positive" from both vertices enter vertex w_{4+5+13}, and the edges labelled "negative" from both vertices enter the sink vertex. Figure 4.7 shows the third step of the backward iteration.

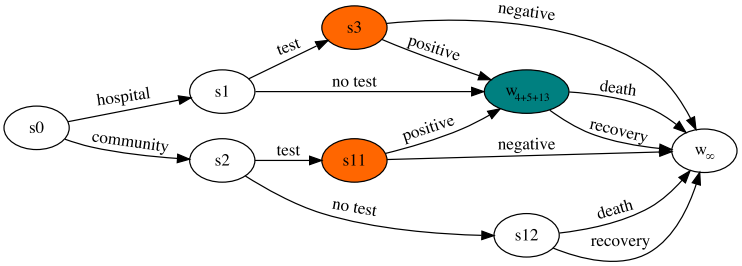


Figure 4.6: Second step of the backward iteration.

Key observation 2: At the kth step of the backward iteration, we consider the situations in the graph of the $(k-1)$th step of the iteration that are at a maximum distance of $k-1$ directed edges from its sink vertex, for $2 \leq k \leq m$. For any two of these situations to be in the same position, the following must hold:

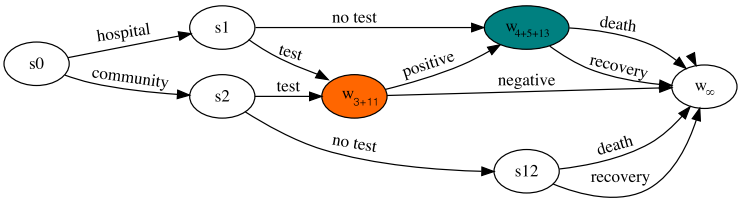


Figure 4.7: Third step of the backward iteration.

- They are members of the same stage set;
- Their emanating edges with identical labels enter the same downstream vertex.

This result is proved in Theorem 4.7.

Example 4.6 (Infection testing example continued)**.** *For the fourth step of the backward iteration we consider situations s_1 and s_2. They satisfy neither of the conditions described in the second key observation. Hence, they are not in the same position and are retained as separate vertices as shown in Figure 4.7. The fifth and final backward step only involves the root vertex. Thus, Figure 4.7 shows the graph of the CEG of this process. Notice that we first constructed this graph in the third step of the iteration.*

Key observation 3: When the graphs obtained from two consecutive iterations are isomorphic, the algorithm can be stopped and the graph thus obtained is the graph of the CEG for the process.

This result is called the *early stopping criterion* and it is proved in Theorem 4.8.

4.3.2 CEG Construction Algorithm

We now formally present our algorithm as reported in Shenvi and Smith (2020a), which given a staged tree $\mathcal{S}$, progressively melds situations together according to the position structure incrementally more distant from the leaves of the staged tree $\mathcal{S}$. This produces a sequence of coloured graphs $\mathcal{G}_0 = \mathcal{S}, \mathcal{G}_1, \ldots, \mathcal{G}_m = C$ where m is the length of the longest root-to-leaf path in $\mathcal{S}$. Each graph in the sequence has the same set of root-to-leaf/sink paths, that is $\mathcal{G}_{0\Lambda} = \mathcal{G}_{1\Lambda} = \ldots = \mathcal{G}_{m\Lambda}$. Note that to describe the algorithm, we redefine path for this subsection to be a sequence of tuples of the form *('vertex colour', 'edge label').*

The following relationship holds between graphs G_i and G_{i+1} in the sequence:

$$|V(G_i)| \geq |V(G_{i+1})|, \quad |E(G_i)| \geq |E(G_{i+1})|; \qquad i = 0, 1, \ldots, m-1.$$

We specify our construction by writing the vertex and edge sets of each graph $\mathcal{G}_i$ as a function of the vertex and edge sets of the graph $\mathcal{G}_{i-1}$. Note that the vertices in $\mathcal{G}_i$ retain their colouring from the graph $\mathcal{G}_{i-1}$. Henceforth, we will say $\mathcal{G}_i = \mathcal{G}_j$, $i \neq j$, whenever the two graphs $\mathcal{G}_i$ and $\mathcal{G}_j$ are isomorphic. Say that a vertex v is at a distance k from the sink vertex w_∞ (or equivalently, a leaf in a tree) if the longest directed path from v to the sink (or a leaf) contains k tuples. Let V^{-k} be the set of vertices in a given graph such that every $v \in V^{-k}$ is at a distance of k from the sink vertex w_∞ (or a leaf) of the graph. We describe our iterative algorithm below.

Step 1: Initialisation. From $\mathcal{G}_0 = \mathcal{S}$ where $\mathcal{S}$ is the staged tree, define the following:

$$\begin{aligned}
&v_1^- \triangleq L(\mathcal{G}_0), \qquad v_1^+ \triangleq \{w_\infty\},\\
&\epsilon_1^- \triangleq \{e \in E(\mathcal{G}_0) \,:\, e = (v, v', l) \text{ where } v \in S(\mathcal{G}_0), v' \in L(\mathcal{G}_0)\},\\
&\epsilon_1^+ \triangleq \{\sigma_1(e) \,:\, e \in \epsilon_1^-\},
\end{aligned}$$

where $\sigma_1(e) = \sigma_1(v, v', l) \triangleq (v, w_\infty, l)$. Graph $\mathcal{G}_1 = (V(\mathcal{G}_1), E(\mathcal{G}_1))$ where

$$V(\mathcal{G}_1) \triangleq V(\mathcal{G}_0) \backslash v_1^- \cup v_1^+, \qquad E(\mathcal{G}_1) \triangleq E(\mathcal{G}_0) \backslash \epsilon_1^- \cup \epsilon_1^+.$$

Step 2: Generalisation. To construct graph $\mathcal{G}_i$ from $\mathcal{G}_{i-1}$, $i \leq m$, proceed as follows:

1. Create a sub-collection $U_i = \{u_{1i}, \ldots, u_{m_i i}\}$ informed by the collection of stages $\mathbb{U}$ such that each situation $v \in V^{-(i-1)}$ belongs to only one set $u_{ji} \in U_i$ for some $j = 1, \ldots, m_i$, and two situations $v, v' \in V^{-(i-1)}$ belong to the same set u_{ji} if and only if there exists a stage $u \in \mathbb{U}$ such that $v, v' \in u$. Thus, the collection U_i gives us the stage structure for the vertices in $V^{-(i-1)}$.

2. Construct a collection U_i^* such that each $u_{ji} \in U_i$ is replaced in U_i^* by the sets $u_{ji}^1, \ldots, u_{ji}^{n_{ji}}$, $n_{ji} \geq 1$. Each situation $v \in u_{ji}$ belongs to only one set u_{ji}^k for some $k = 1, \ldots, n_{ji}$, and two situations $v, v' \in u_{ji}$ belong to the same set u_{ji}^k if and only if there exists an edge $(v', v'', l) \in E(\mathcal{G}_{i-1})$ for every edge $(v, v'', l) \in E(\mathcal{G}_{i-1})$. Thus, we have that $u_{ji}^k \cap u_{ji}^l = \emptyset$, $k \neq l$, $\cup_k u_{ji}^k = u_{ji}$, and $U_i^* = \cup_j \cup_k u_{ji}^k$. The collection U_i^* partitions the situations in $V^{-(i-1)}$ into its position sets (see key observation 2).

3. Define the following terms for each u_{ji}^k, $j = 1, \ldots, m_i$, $k = 1, \ldots, n_{ji}$,

$$v^-(u_{ji}^k) \triangleq u_{ji}^k, \qquad v^+(u_{ji}^k) \triangleq \{v\} \text{ for some } v \in v^-(u_{ji}^k).$$

The vertices in $\nu^-(u^k_{ji})$ are contracted into a single vertex represented by $\nu^+(u^k_{ji})$ in $\mathcal{G}_i$. We now define the following terms to enable us to construct the vertex and edge sets of $\mathcal{G}_i$,

$$\begin{aligned}
&\nu_i^- \triangleq \cup_j \cup_k \nu^-(u^k_{ji}), \qquad \nu_i^+ \triangleq \cup_j \cup_k \nu^+(u^k_{ji}),\\
&\epsilon_i^f \triangleq \{e \in E(\mathcal{G}_{i-1}) \,:\, e = (v, v', l) \text{ where } v \in \nu_i^- \backslash \nu_i^+\},\\
&\epsilon_i^b \triangleq \{e \in E(\mathcal{G}_{i-1}) \,:\, e = (v, v', l) \text{ where } v' \in \nu_i^- \backslash \nu_i^+\},\\
&\epsilon_i^- \triangleq \epsilon_i^f \cup \epsilon_i^b, \qquad \epsilon_i^+ \triangleq \{\sigma_i(e) \,:\, e \in \epsilon_i^b\},
\end{aligned}$$

where $\sigma_i(e) = \sigma_i(v, v', l) \triangleq (v, v'', l)$ in which $v'' \in \nu^+(u^k_{ji})$ for $v' \in \nu^-(u^k_{ji})$, $k = 1, \ldots, n_{ji}$. Setting $V(\mathcal{G}_i) \triangleq V(\mathcal{G}_{i-1}) \backslash \nu_i^- \cup \nu_i^+$ and $E(\mathcal{G}_i) \triangleq E(\mathcal{G}_{i-1}) \backslash \epsilon_i^- \cup \epsilon_i^+$ gives us the graph of $\mathcal{G}_i$

We now prove as Theorem 4.7 that the above construction of U_i^* does in fact result in a collection of positions of the vertices in $V^{-(i-1)}$.

Theorem 4.7. *Given graph $\mathcal{G}_{i-1}, i \leq m$ in the sequence of graphs transforming a staged tree $\mathcal{G}_0 = \mathcal{S}$ to a CEG $\mathcal{G}_m = C$, two situations $v_1, v_2 \in V^{-(i-1)}$ are in the same position if and only if they belong to the same stage and for every (v_1, v', l) there exists a (v_2, v', l) in $\mathcal{G}_{i-1}$.*

Proof. As the base case, consider the situations in V^{-1} in $\mathcal{G}_1$ where the leaves of $\mathcal{S}$ are contracted into the sink vertex. Situations in V^{-1} are necessarily in the same position whenever they are in the same stage as their rooted subtrees in $\mathcal{S}$ are florets. Additionally, all their emanating edges terminate in the sink vertex.

To generalise, consider now the graph $\mathcal{G}_{i-1}$ belonging to the sequence of graphs converting a staged tree $\mathcal{S}$ into a CEG C, for $i = 3, \ldots, m$. All the vertices in V^{-j}, $j = 1, \ldots, i-2$ in $\mathcal{G}_{i-1}$ represent positions.

$\Rightarrow$ Given that two situations $v_1, v_2 \in V^{-(i-1)}$ are in the same position. We show that (1) v_1 and v_2 belong to the same stage; (2) for every (v_1, v', l) there exists a (v_2, v', l) in $\mathcal{G}_{i-1}$.

If v_1 and v_2 are in the same position, it is trivially true that they are also in the same stage. Additionally, by the definition of a position, the subtrees rooted at v_1 and v_2, call them $\mathcal{S}_{v_1}$ and $\mathcal{S}_{v_2}$ in the staged tree $\mathcal{S}$ are isomorphic. Thus also, for every subtree rooted at a child of v_1 in $\mathcal{S}_{v_1}$, there exists an isomorphic subtree rooted at a child of v_2 in $\mathcal{S}_{v_2}$. In fact, stages by definition require that edges with the same estimated conditional transition probability must also have the same edge label. Therefore, there necessarily exists a situation v_2^{ch} along edge (v_2, v_2^{ch}, l) such that the subtree rooted at v_2^{ch} is isomorphic to the subtree rooted at situation v_1^{ch} which is along the edge (v_1, v_1^{ch}, l). Notice that v_1^{ch} and v_2^{ch} belong to the set $V^{-(i-2)}$ in $\mathcal{G}_{i-2}$. Since their rooted subtrees in $\mathcal{S}$ are isomorphic, they belong to the same

position and are represented by a single vertex, say $v^{ch}_{1,2}$ in $\mathcal{G}_{i-1}$. The edges (v_1, v^{ch}_1, l) in $\mathcal{S}_{v_1}$ and (v_2, v^{ch}_2, l) in $\mathcal{S}_{v_2}$ are represented by edges $(v_1, v^{ch}_{1,2}, l)$ and $(v_2, v^{ch}_{1,2}, l)$ in $\mathcal{G}_{i-1}$. This result extends to every (v_1, v', l) in $\mathcal{G}_{i-1}$.

$\Leftarrow$ Given that $v_1, v_2 \in V^{-(i-1)}$ in $\mathcal{G}_{i-1}$ belong to the same stage and for every (v_1, v', l) there exists a (v_2, v', l) in $\mathcal{G}_{i-1}$. We need to show that v_1 and v_2 are in the same position.

Recall that two situations are in the same position when the subtrees rooted at these vertices in $\mathcal{S}$ are isomorphic. Since v_1 and v_2 are in the same stage, they have the same number of emanating edges and also, the edges from v_1 and v_2 which share the same edge label have the same estimated conditional transition probability. Consider edges $(v_1, v^{ch}_{1,2}, l)$ and $(v_2, v^{ch}_{1,2}, l)$ emanating from situations v_1 and v_2 in $\mathcal{G}_{i-1}$ respectively where $v^{ch}_{1,2}$ is the common situation along these two edges. In a tree each vertex has at most one parent. So in the staged tree $\mathcal{S}$, the position $v^{ch}_{1,2}$ would be represented by two separate vertices, call them v^{ch}_1 and v^{ch}_2 in the subtrees rooted at v_1 and v_2 respectively. Thus, the edge $(v_1, v^{ch}_{1,2}, l)$ would be replaced by an edge (v_1, v^{ch}_1, l) in the subtree rooted at v_1, call this $\mathcal{S}_{v_1}$ in $\mathcal{S}$. Similarly, the edge $(v_2, v^{ch}_{1,2}, l)$ would be replaced by an edge (v_2, v^{ch}_2, l) in $\mathcal{S}_{v_2}$ which is the subtree rooted at v_2 in $\mathcal{S}$. Since v^{ch}_1 and v^{ch}_2 are in the same position in $\mathcal{G}_{i-1}$, they have isomorphic subtrees in $\mathcal{S}_{v_1}$ and $\mathcal{S}_{v_2}$. Similarly, the subtrees rooted at the other children of v_1 and v_2 in $\mathcal{S}_{v_1}$ and $\mathcal{S}_{v_2}$ respectively are isomorphic whenever the edges from v_1 and v_2 to their respective children share the same edge label. Since v_1 and v_2 are in the same stage, the florets $F(v_1)$ in $\mathcal{S}_{v_1}$ and $F(v_2)$ in $\mathcal{S}_{v_2}$ are also isomorphic. Thus $\mathcal{S}_{v_1}$ and $\mathcal{S}_{v_2}$ are isomorphic and hence, they belong to the same position. □

We now show that the recursion may in fact be stopped for some $0 < r < m$. This optimal stopping criterion is presented below in Theorem 4.8 with its proof.

Theorem 4.8 (Optimal Stopping Criterion)**.** *In the sequence of graphs transforming a staged tree $\mathcal{G}_0 = \mathcal{S}$ to a CEG $\mathcal{G}_m = C$ and $m \geq 2$ where m is the depth of $\mathcal{S}$, the earliest stopping time in this transformation that guarantees the required CEG C is the recursion step r such that $\mathcal{G}_r = \mathcal{G}_{r-1} \neq \mathcal{G}_{r-2}$, $0 < r < m$.*

Proof. Suppose that $0 < r < m$ recursions have taken place and $\mathcal{G}_r = \mathcal{G}_{r-1} \neq \mathcal{G}_{r-2}$. First we show that $\mathcal{G}_r = C$. As the graph of a CEG is the most parsimonious representation of the event tree describing a process, this is equivalent to showing that $|V(\mathcal{G}_r)| = |\mathbb{W}| + 1$ where $\mathbb{W}$ is the collection of positions. Graph $\mathcal{G}_{r-1}$ contains the positions for all situations in V^{-k}, $k < r - 1$ (from Theorem 4.7). Thus the problem can be framed as showing that if there are no non-trivial positions in $V^{-(r-1)}$ then there are no non-trivial positions in any of V^{-k}, $r \leq k \leq m$. We prove this by contradiction.

Let there be no non-trivial positions in $V^{-(r-1)}$. Suppose that two situations $v_1, v_2 \in V^{-r}$ are in the same position and hence, the same stage. This implies that the subtrees of

$\mathcal{S}$ rooted at v_1 and v_2, say $\mathcal{S}_{v_1}$ and $\mathcal{S}_{v_2}$ respectively are isomorphic. Let v_1^{ch} be a child of v_1 along the edge (v_1, v_1^{ch}, l) and let $\mathcal{S}_{v_1^{ch}}$ be the subtree rooted at v_1^{ch}. By the definition of a stage, there exists an edge (v_2, v_2^{ch}, l) in $\mathcal{S}_{v_2}$ with rooted subtree $\mathcal{S}_{v_2^{ch}}$. The subtrees $\mathcal{S}_{v_1^{ch}}$ and $\mathcal{S}_{v_2^{ch}}$ are isomorphic as $\mathcal{S}_{v_1}$ and $\mathcal{S}_{v_2}$ are isomorphic. By the definition of a position, v_1^{ch} and v_2^{ch} are in the same position. As $v_1, v_2 \in V^{-r}$, we have that $v_1^{ch}, v_2^{ch} \in V^{-(r-1)}$. This contradicts that there are no non-trivial positions in $V^{-(r-1)}$. A similar argument can be made for any $v_1, v_2 \in V^{-k}$, $r \leq k \leq m$. Since $\mathcal{G}_r = \mathcal{G}_{r-1}$, $V^{-(r-1)}$ has no non-trivial positions and all the positions in V^{-k}, $k < r$ have been identified. By the above result, V^{-k}, $r \leq k \leq m$ also do not contain any non-trivial positions. Thus $\mathcal{G}_r = C$.

We have that $\mathcal{G}_{r-2} \neq \mathcal{G}_{r-1} = \mathcal{G}_r = \ldots = \mathcal{G}_m = C$. While stopping at graph G_{r-1} gives us the required graph of the CEG, this recursive step is indistinguishable from any of the other $k < r - 1$ steps. Hence, the isomorphism of $\mathcal{G}_{r-1}$ and $\mathcal{G}_r$ is needed to stop the recursions with certainty. Thus the earliest stopping point for the recursion is step r such that $\mathcal{G}_r = \mathcal{G}_{r-1} \neq \mathcal{G}_{r-2}$, $0 < r < m$. □

Theorem 4.9 implies that for every staged tree there is a unique CEG and also that the staged tree can be recovered given this CEG. This is equivalent to saying that no information is lost in transforming a staged tree into a CEG. Research on CEGs has always assumed this to be true but a proof for this foundational assumption had been missing from the literature.

Theorem 4.9 (Preservation of Information)**.** *The mapping from a staged tree to its CEG is bijective.*

Proof. We prove bijection by proving injection and surjection.

Injection: We prove the injective contrapositive; i.e. given staged trees $\mathcal{S}_1 \neq \mathcal{S}_2$, we show that their corresponding CEGs C_1 and C_2 are not isomorphic. It is straightforward to show that if $\mathcal{S}_1$ and $\mathcal{S}_2$ are structurally not isomorphic, then $C_1 \neq C_2$. Suppose that $\mathcal{S}_1$ and $\mathcal{S}_2$ are structurally isomorphic and that they differ only in the colouring of one of their vertices. Let these vertices be v_1 with colour c_1 in $\mathcal{S}_1$ and v_2 with colour $c_2 \neq c_1$ in $\mathcal{S}_2$. Since vertices retain their colouring in the CEG, the positions representing v_1 and v_2 in C_1 and C_2 will be coloured by c_1 and c_2 respectively. Hence, C_1 and C_2 will not have colour preserving isomorphism. Additionally, C_1 and C_2 will not be structurally isomorphic if either or both of v_1 and v_2 create non-trivial positions in their respective staged trees as the collection of positions in $\mathcal{S}_1$ and $\mathcal{S}_2$ will not be equivalent.

Surjection: From a given CEG C, construct a staged tree $\mathcal{S}$ as follows:

1. Sort the root-to-sink paths in C_Λ in ascending order of the length (number of tuples) of the paths.

2. For each path of length $\ell \geq 1$, let $\{(c, l)\}$ denote the first tuple in the path where c is its colour and l is its edge label. Construct an edge from v_0, the root of $\mathcal{S}$ to a new vertex (labelled as v_i where i is an integer index which hasn't been assigned thus far in the construction) and label it l. Assign colour c to v_0.

3. In general, for any path of length $\ell \geq k$ for $k = 2, 3, \ldots, m$, let the sequence up to its kth tuple be given by $\{(c_1, l_1), \ldots, (c_{k-1}, l_{k-1}), (c_k, l_k)\}$. There necessarily exists a path given by the sequence $\{(c_1, l_1), \ldots, (c_{k-1}, l_{k-1})\}$ ending in a vertex, say v_{k-1} in the staged tree constructed so far. To add the kth tuple (c_k, l_k) to this path, colour v_{k-1} by c_k, add a vertex v_k and construct a directed edge from v_{k-1} to v_k with edge label l_k.

This construction results in a tree as it is connected (no vertex – with the exception of the root – is added until it is connected by an edge to an existing vertex) and has no directed cycles (each edge is constructed from an existing vertex to a new vertex). Call this tree $\mathcal{T}^*$. We prove that $\mathcal{T}^*$ is the unique staged tree whose transformation, as given in our algorithm, results in our given CEG C.

Observe that a staged tree in uniquely and unambiguously defined by its underlying event tree and its collection of stages $\mathbb{U}$. The structure of any event tree can be recovered from its set of uncoloured root-to-leaf paths, which is equivalent to the uncoloured root-to-sink paths C_Λ of the CEG C. As $\mathcal{T}^*$ is constructed from the set C_Λ, the uncoloured version of $\mathcal{T}^*$ is the required underlying event tree for C. The vertices of $\mathcal{T}^*$ inherit their colourings from the vertices of C. Recall that colouring of vertices in a CEG is indicative of stage memberships. Hence, two vertices w and w' with the same colour, say c in C are in the same stage. By definition of a stage, $\boldsymbol{\theta}_w = \boldsymbol{\theta}_{w'}$, and for each edge $e = (w, \cdot, l)$ there exists an edge $e' = (w', \cdot, l)$ such that $\theta(e) = \theta(e')$ in C. Two vertices v and v' with the colour c in $\mathcal{T}^*$ either belong to the same position set in C – without loss of generality assume this position set is represented by w – or belong to two distinct position sets in C, assume these are represented by w and w'. If both v, v' belong to position set represented by w, then v and v' in $\mathcal{T}^*$ are created from two separate root-to-w subpaths, say p and p' in C. Floret $F(v)$ is formed by creating k copies of subpath p and appending each with a distinct (c, l_i) where $i = 1, \ldots, k$ and l_i is the label of the ith edge emanating from w in C. Floret $F(v')$ is constructed in a similar manner. Thus v and v' have the same number of emanating vertices in $\mathcal{T}^*$ and share the same vertex colour as they satisfy the conditions of being in the same stage by belonging to the same position in C. This also holds when v and v' belong to position sets represented by w and w' respectively, where w and w' share the same colour in C, with the exception that p will be a root-to-w subpath and p' a root-to-w' subpath. Thus $\mathcal{T}^*$ is the underlying staged tree of C as it has the structure of the event tree of C and a collection of stages equivalent to that of C. □

4.3.3 Related Work

An algorithm, similar to the one presented in Section 4.3.2 above, was presented in Silander and Leong (2013) to learn a stratified staged tree from its underlying event tree and to transform it into a stratified CEG (although the stratified terminology was not explicitly used). The input for their algorithm is the staged tree of an $\boldsymbol{X}(I)$-compatible event tree for some $\boldsymbol{X} = \{X_1, X_2, \ldots, X_n\}$ and some permutation I. Such a staged tree has $n + 1$ layers; n layers where each layer contains situations representing the same variable, and a final layer representing the leaves of the staged tree. Their algorithm involves a backward iteration through all the layers to identify situations belonging to the same position. In their paper, the term "layer" is not directly defined and this ambiguity needs to be resolved before their algorithm can be applied to a non-stratified staged tree. By defining each layer k to be equivalent to the set V^{-k} (defined in Section 4.3.2) $k = 0, 1, \ldots, n$, their algorithm can be applied to non-stratified staged trees. However, the algorithm presented in Silander and Leong (2013) involves certain non-trivial steps (analogous to Theorem 4.7) for which they provided no proof. Additionally, as they did not provide an early stopping criterion, their algorithm requires iterating through all the layers of the staged tree.

4.3.4 Experiments for the CEG Construction Algorithm

We now examine how our staged tree to CEG construction algorithm performs when compared to an adapted version of the algorithm presented by Silander and Leong (2013).

We adapt the algorithm in Silander and Leong (2013) so that layer k in their algorithm corresponds to what we defined as set V^{-k} in Section 4.3.2. While we also proved the theoretical results associated with the algorithm, for practical purposes the difference between the adapted version of the algorithm in Silander and Leong (2013) and our algorithm is that ours comes with an early stopping criterion which allows the iteration to be stopped earlier when certain conditions are met. For convenience, call their adapted algorithm the *baseline algorithm* and ours the *optimal time algorithm*.

We compare the performance of the baseline and optimal time algorithms on 7 datasets. The collection of stages in the event tree of the process are first identified using the AHC algorithm. The performance of the baseline and optimal time algorithms are then compared based on their runtimes to construct the CEG from the staged tree obtained using the AHC algorithm.

The first four datasets used for the comparison are from the UCI repository (Dua & Graff, 2019). The missing values were removed and sampling zeros were treated as structural for simplicity. The fifth dataset is from the Christchurch Health and Development Study (CHDS) conducted at the University of Otago, New Zealand (see Fergusson et al.

(1986)). The penultimate dataset is simulated data for the falls intervention as presented in Section 4.5. The final dataset is an extension of this dataset presented in Section 5.7. The last two datasets have structural asymmetries. Thereby their event trees are not stratified and the atoms of their event spaces given by their root-to-leaf paths are not all of the same length. The event trees for the remaining datasets are stratified.

These experiments were run on a 2.9 GHz MacBook Pro with 32GB memory using my Python code https://github.com/ashenvi10/Chain-Event-Graphs. This code can handle datasets with structural asymmetries (stored as NaNs or null values) and also provides the capability to manually add sampling zero paths to the tree. The code is currently set up to learn the staged tree from the event tree of the dataset using the AHC algorithm with hyperstages (defined in Section 4.4) being an optional argument. Observe that, unlike the existing 'ceg' (Collazo & Taranti, 2017) and 'stagedtrees' (Carli et al., 2020) R packages, our code is not restricted to stratified CEGs and it also allows manual addition of edges with sampling zeros.

Dataset	$\lvert\mathcal{S}\rvert$	**Depth** m	$\lvert V(C)\rvert$	T_{Baseline}	T_{Optimal}
Iris	52	5	42	1.635	1.414
Hayes-Roth	124	5	58	12.118	12.085
Balance scale	327	5	90	145.052	143.321
Glass	636	10	308	389.272	376.689
CHDS	19	4	10	0.586	0.556
Falls	39	6	27	1.564	1.453
Falls dynamic	346	5	242	585.789	550.990

Table 4.1: Comparison of the baseline algorithm and the optimal time algorithm.

Table 4.1 gives for each dataset the number of situations in the staged tree output by the AHC algorithm ($|\mathcal{S}|$), the maximum depth of the staged tree (m), the number of positions in the resulting CEG found by the two algorithms ($|V(C)|$) and the time taken (in milliseconds) by the two compacting algorithms (T_{Baseline} and T_{Optimal}). Due to difficulty scaling the AHC algorithm, we allowed situations to be in the same stage only if they satisfied the additional condition of being at the same distance from some leaf of the staged tree. However, this does not affect the performance of the baseline or optimal time algorithm as situations can only be in the same position if they are at the same distance from some leaf.

From Table 4.1 we can see that the optimal time algorithm takes less time than the baseline algorithm while arriving at the same CEG as it stops as soon as Theorem 4.8 is satisfied. The time saved using the optimal algorithm compared to the baseline algorithm is quite modest in the datasets considered here. This is because the gain in efficiencies are inversely proportional to the farthest distance of a situation, belonging to a non-trivial

position set in the staged tree, from a leaf of the staged tree. For instance, if all situations belonging to non-trivial position sets are situated in V^{-1} then using the optimal stopping criterion, the iteration can be stopped after obtaining the intermediate graph $\mathcal{G}_3$ and this saves us searching through V^{-i}, $i = 3, 4, \ldots, n$. Whereas, if there were situations in $V^{-(n-1)}$ belonging to the same position set, then there is no gain in efficiency between the baseline and optimal algorithms.

4.4 Model Selection for Non-Stratified CEGs

Conjugate learning and prior specification for non-stratified CEGs proceeds exactly as described for stratified CEGs in Chapter 3. In this section, we discuss how the model selection algorithms described in Section 3.2.3 can be extended to search the model space of non-stratified CEGs.

The model search space of the non-stratified CEG class is very large. A full search through this space quickly becomes infeasible as the number of events represented by the event tree increases. Hence, to enable a meaningful search through this vast space, we introduce the concepts of *square-free staged trees and CEGs* (Collazo et al., 2018), and *hyperstages* (Collazo, 2017).

Definition 4.10 (Square-free staged tree (/CEG))**.** *A staged tree $\mathcal{S}$ (/CEG C) is said to be square-free if no two situations that lie on the same root-to-leaf (/root-to-sink) path are in the same stage set.*

We assume here that the staged trees and hence the CEGs we consider in this chapter are *square-free*. For vanilla CEGs, the square-free requirement is a natural one. It is generally of interest to know whether the one-step evolution of two situations is equivalent only when they represent the same type of event. While our definition of stages tries to ensure that this is the case by requiring that the edge labels and their corresponding transition probabilities are equivalent between situations in the same stage, it is possible that inherently different events have the same edge labels. For example, in the falls intervention, both X_{Re} (referral) and X_F (fall status), could have outgoing edges with labels "Yes" and "No".

There are, of course, scenarios where such an assumption may not be appropriate. For example, in the modified falls example, it could be of interest to identify whether any situations representing variables X_{Ri} (risk) and X_{Rev} (revised risk) are in the same stage since they both represent similar events. For such processes, we can assume that the associated event tree and CEG are not square-free. The concept of hyperstages described below may still be used in these processes to narrow the search space.

Definition 4.11 (Hyperstage)**.** *A hyperstage* $\boldsymbol{H} = \{H_1, H_2, \ldots, H_n\}$ *for the situations* $S(\mathcal{T})$ *of an event tree* $\mathcal{T}$ *is a collection of sets such that any two situations* v *and* v' *can be in the same stage in* $\mathbb{U}$ *only if there exists a set* $H_i \in \boldsymbol{H}$ *such that* $v, v' \in H_i$*.*

Observe that the definition of a hyperstage does not require the sets contained within the hyperstage to be mutually exclusive. However, the model selection process is greatly simplified when these sets are mutually exclusive. Within this thesis, we shall assume that the sets contained within a hyperstage are mutually exclusive. The hyperstage enables the modeller and domain experts to encode sets of structural and causal hypotheses into the process of model selection (Collazo, 2017). In fact, the assumption of a staged tree or equivalently, its associated CEG being square-free is encoded within the hyperstage defined for the process.

We now discuss how the model selection algorithms described in Section 3.2.3 for stratified CEGs can be extended to non-stratified CEGs using the concept of a hyperstage. In theory, the AHC algorithm for model selection can be directly applied to the non-stratified class as it does not have any constraints specific to the stratified class. Recall that the AHC is a greedy model search algorithm and it checks for score improvements obtained by pairwise merging of every pair of stages that satisfy the requirement of having the same number of outgoing edges and the same set of edge labels. However, inherently different situations can have equivalent sets of edge labels. For the stratified class, this problem can be overcome by running the AHC algorithm on each layer of a stratified event tree as described in Section 3.2.3. By definition, each layer of the stratified event tree represents the same set of events. We can replicate this strategy within the non-stratified class by constructing its hyperstage such that each set within the hyperstage contains situations that satisfy both the above stated conditions and also represent the same types of events. In this case, each set of the hyperstage for the non-stratified event tree is analogous to each layer of the stratified event tree.

Example 4.12 (Falls intervention example)**.** *For instance, for the event tree in Figure 4.1, the hyperstage could be given by*

$$\begin{aligned}\boldsymbol{H} = \{&\{s_0\}, \{s_1, s_2\}, \{s_3\}, \{s_4, s_5, s_6\}, \{s_7\}, \{s_8, s_9\}, \{s_{10}, s_{11}\},\\ &\{s_{12}, s_{13}, s_{14}, s_{15}, s_{16}, s_{17}, s_{18}\}\}.\end{aligned}$$

When the hyperstage partitions the vertex set of an event tree, running the AHC algorithm on the entire event tree can be simplified to running AHC independently on each of the sets in the hyperstage. However if the sets in the hyperstage are not mutually exclusive, we cannot optimise by identifying the stages in each set of the hyperstage independently

and the AHC algorithm must be run over the entire event tree. The search space is still restricted due to the hyperstage.

For mutually exclusive sets in the hyperstage, the dynamic programming approach to model selection can be adopted to find the best stage partition within each set of the hyperstage. However, dynamic programming cannot be implemented when the sets of the hyperstage are not mutually exclusive. In Section 3.2.3 we discussed how the dynamic programming approach can also be used to find the best ordering of the variables describing the event tree. This approach cannot be directly used for the non-stratified class as the ordering of the variables need not be the same across all root-to-leaf paths of the event tree. This leads to an extremely large search space even for a small problem. However, restrictions can be introduced based on the application to reduce the search space. For instance, a non-strict total ordering of the variables would vastly reduce the search space.

Finally, we caution that when performing model selection for the non-stratified class, if the event tree is being constructed directly from data then care must be taken to not assume that all occurrences of zero edge counts are structural zeros. The dataset could have sampling zeros which are occurrences of zero observations due to sampling limitations as discussed in Section 4.1. In such cases, any edges with sampling zeros would need to be added to the event tree.

4.5 Application of the Falls Intervention

In this section, we analyse a simulated dataset based on the extension of the falls intervention (first described in Section 4.1) as presented in Shenvi et al. (2018). We then compare the use of a BN and a CEG to model this public health intervention. Here we additionally classify individuals by their type of residence as the domain literature suggests that the fall rates are higher for individuals living in institutionalised care. Below we describe the domain information used to simulate our dataset.

2011 Census: Aggregate Data (2011) distinguishes the usual residence (living for six months or longer) of individuals as community-dwelling or living in communal establishments (as defined in "Office for National Statistics: 2011 Census Glossary" (2011)). Communal establishments include care homes, nursing homes and hospitals. Close to 96.3% of those aged over 65 in England and Wales lived in the community in 2011. For our data simulation, we assume this proportion has not changed drastically in the years since. The risk of falling for those living in communal establishments is significantly higher than for those living in the community which could be due to the reduced general health and increased frailty of those who tend to live in nursing homes and hospitals (Cameron et al., 2010). It was found that the falls incidence in nursing homes is roughly thrice that in the

community with 1.4 falls per person per year for residents of a nursing home (Rubenstein et al., 1994; Nurmi & Lüthje, 2002). The fall rates in hospitals vary between wards and in-hospital fall records show rates from 2.9-13 falls per 1,000 bed days (Morse & Field, 1996).

2011 Census: Aggregate Data (2011) provides details of the self-reported health status of those aged over 65 years in England and Wales. Individuals were asked to classify their health as "Very good or good health", "Fair health" or "Bad or very bad health". This information is summarised in Table 4.2. We assume that the proportion of high risk individuals is equivalent to the proportion of individuals who self-reported their health status as "Bad or very bad health". Under this assumption, roughly 36.35% of individuals in communal establishments and 14.74% of individuals in the community are assumed to be at a high risk of falling. These figures match with the overall proportion of high risk individuals in the population as found in Eldridge et al. (2005) and with the risk of falling for individuals in nursing homes as found in Nurmi and Lüthje (2002) adjusted for removal of recurrent falls. Further, based on the referral pathways in Eldridge et al. (2005), we assume that it was more likely for the intervention to assess a greater proportion of individuals living in communal establishments (we set this at 20%) than in the community (we set this at 6%). We also assume that a higher proportion of high risk communal establishment individuals would be assessed by the intervention than the corresponding proportion for community dwellers due to the design of the assessment pathways in the intervention.

	Communal Establishments	Community
Very good or good health	50,058	4,473,000
Fair health	146,409	3,102,886
Bad or very bad health	112,210	1,309,752
Total	308,677	8,885,638

Table 4.2: Self-reported health status of individuals aged over 65 living in England and Wales separated by type of usual residence.

The conditional transition probabilities for the data generating event tree were determined by the above assumptions. The data generating staged tree is given in Figure 4.8. The vertex labels for the leaf vertices are hidden to prevent visual cluttering. Note here that we combine the variables for the residence type and assessment into one. Similarly, we combine the variables for referral and treatment into one. This does not affect the inference or the reading of conditional independences from the topology of the resultant CEG as the probability of assessment, referral and treatment are set by intervention design. Additionally, here we treat the combined variable of referral and treatment as having no logical interpretation for individuals who have not been assessed, as by intervention design they do

not receive any referral or treatment. Hence we treat it here as structurally missing rather than as a structural zero with no effect on the inference. For this illustration we generate a dataset of 50,000 individuals by forward sampling. The numbers along the edges in Figure 4.8 represent the observations along each edge. Observe that several of the branches are sparsely populated. For instance, there are only two observations along the edge indicating falls suffered by assessed low risk individuals in communal establishments who received treatment. Sparsely populated edges may pose a problem for model selection. We discuss this further in Section 4.6.

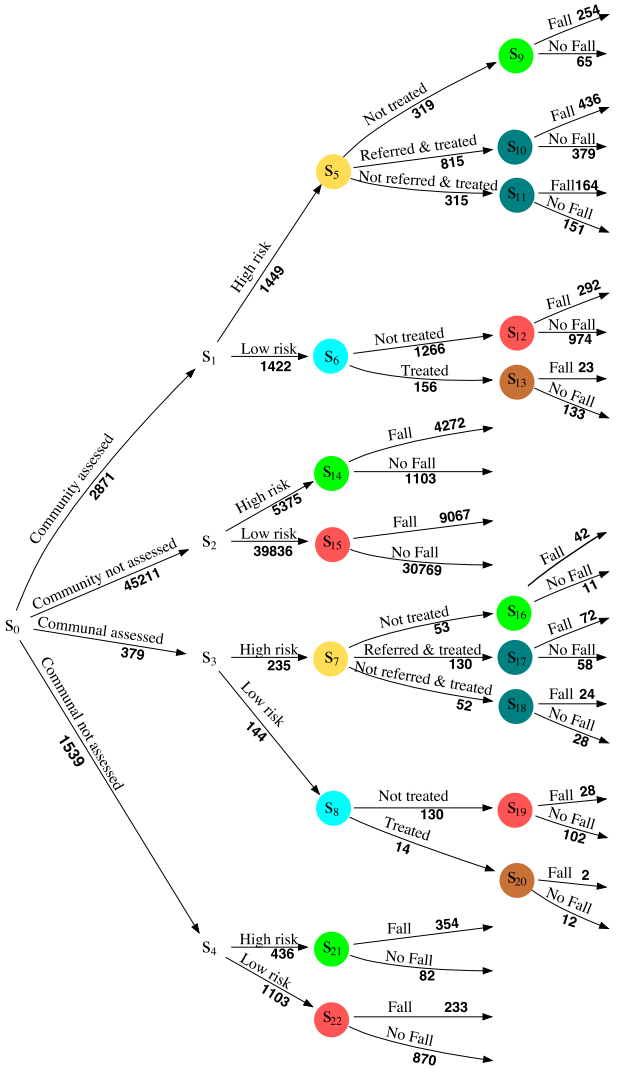


Figure 4.8: Staged tree representing the data generating model.

Here we model this modified falls intervention with a CEG as well as a BN. We can define the variables used to study this intervention as $\boldsymbol{X''} = \{X_A, X_{Ri}, X_T, X_F\}$. Here X_A indicates whether the individual aged over 65 resides in the community or in a communal establishment (such as nursing homes, care homes, hospitals) and whether they have been assessed or not, and X_T indicates whether the individual has been referred & treated, not referred & treated or not treated. X_{Ri} and X_F are defined as before.

For identifying the MAP CEG, we specify the hyperparameters using the property of mass conservation as described in Section 3.2.2 with a weakly informative imaginary sample size ($\tilde{\boldsymbol{\alpha}}_0$) of 4. All CEG structures are assumed *a priori* equally likely. The hyper-stage is given by

$$\mathbf{H} = \{\{s_o\}, \{s_1, s_2, s_3, s_4\}, \{s_5, s_7\}, \{s_6, s_8\}, \{s_9, s_{10}, s_{11}, s_{14}, s_{16}, s_{17}, s_{18}, s_{21}\}, \{s_{12}, s_{13}, s_{15}, s_{19}, s_{20}, s_{22}\}\}.$$

The MAP CEG returned by the AHC algorithm is given in Figure 4.9. Note that the stage structure of this CEG is equivalent to the stage structure of the data generating tree given in Figure 4.8. The log marginal likelihood score of this CEG is -68,671.59.

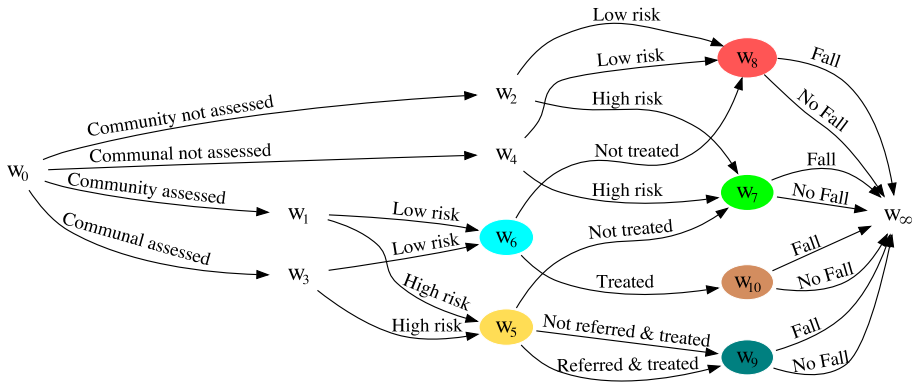


Figure 4.9: MAP CEG returned by the AHC algorithm.

To learn the BN structure, we use the Hill-Climbing algorithm from the R package 'bnlearn' (Scutari, 2010) with the BD equivalent uniform metric to compare BN structures. The Hill-Climbing algorithm outputs the BN in Figure 4.10(a). As the intervention naturally gives rise to a total order of $X_A \prec X_{Ri} \prec X_T \prec X_F$, we suppress certain edges in order for the BN to be representative of our application. For instance, the directed edge from *Treatment* to *Risk* is suppressed given the total order. This gives rise to the BN in Figure 4.10(b). The log marginal likelihood score of this BN structure is -68,709.99.

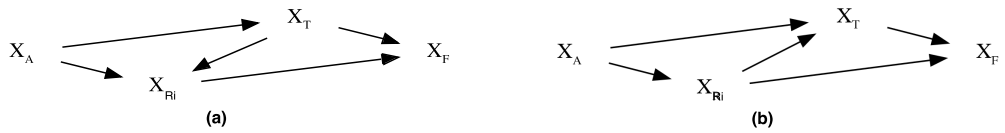


Figure 4.10: (a) Original BN returned using the Hill-Climbing algorithm; (b) Best-fitting BN which admits the total order of $X_A \prec X_{Ri} \prec X_T \prec X_F$.

Thus the log Bayes Factor (Bayes Factor) against the BN model is 38.4 (4.7523×10^{16}) which may be interpreted as very strong evidence against the BN model when compared to the CEG (see Section 3.2.1). Additionally, from the topology of the graph of the MAP CEG in Figure 4.9, we can directly see that the variable of treatment X_T is not defined for both risk groups and both residential settings for those who are not assessed. Whereas, variable X_T is defined for those who are assessed and live in either residential setting, irrespective of their risk group. This information cannot be inferred from the graph of the BN in Figure 4.10(b).

We now perform sensitivity analysis on the choice of the root imaginary sample size $\bar{\boldsymbol{\alpha}}_0$. We learn the MAP CEG from the data using the AHC algorithm with varying values of $\bar{\boldsymbol{\alpha}}_0$ and compare the number of stages at each value of $\bar{\boldsymbol{\alpha}}_0$ between 0.25 and 20 with increments of 0.25 as shown in Figure 4.11. The number of stages and in fact, the stage structure for $\bar{\boldsymbol{\alpha}}_0$ greater than three are equivalent to the data generating model.

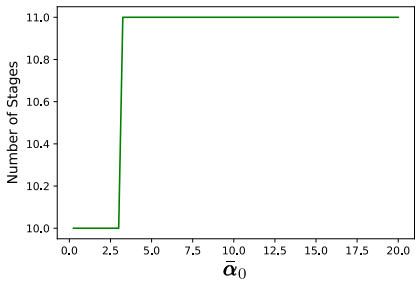


Figure 4.11: The number of stages in the MAP CEG model of the falls intervention for varying values of $\bar{\boldsymbol{\alpha}}_0$.

Finally, we note here CEGs admit exploration of causal hypotheses through manip-

ulations under certain conditions as described in detail in Thwaites et al. (2010), Barclay et al. (2013), and Thwaites (2013). Such CEGs are called causal CEGs. Manipulations in CEGs can be asymmetric as it is possible to intervene in certain positions and not necessarily on the entire variable. For instance, assuming our non-stratified CEG in Figure 4.9 is also a causal CEG, we may wish to examine the effect of referral and treatment given to all assessed high risk individuals irrespective of their type of residence. This would result in the deletion of edge w_5 to w_7 as well as the edge labelled "not referred & treated" from w_5 to w_9. The edge probabilities remain unchanged except that the probability of traversing the remaining w_5 to w_9 edge conditional on reaching w_5 is one.

4.6 Conclusion

We presented a simple iterative backward algorithm along with supporting Python code to transform any staged tree into a CEG. Research in CEGs and their applications has been an increasingly active field in recent years. However, such a general algorithm and proofs of the validity of the staged tree to CEG transformation had been missing in the literature so far. A soon to be published d-separation theorem for CEGs has been developed (Wilkerson, 2020). Construction of ancestral CEGs in this theorem follows the same procedure as our algorithm. Hence, automating this process, as we have done, is a very timely development. Further, by providing the associated Python code accommodating stratified and non-stratified CEGs, we hope to motivate further applications using non-stratified CEGs.

We observed for our modified falls intervention example in Section 4.5 that the CEG appears to be robust against varying values of $\bar{\boldsymbol{\alpha}}_0$. For $\bar{\boldsymbol{\alpha}}_0$ greater than three, the resultant CEG was the data generating model. The smaller values of $\bar{\boldsymbol{\alpha}}_0$ were unable to return this perhaps due to sparse observations along some of the edges of the event tree. In the falls intervention scenario we had domain literature to support the veracity of the staging produced by the AHC. It has been shown in Collazo and Smith (2016) that the AHC algorithm with local Dirichlet priors may merge stages without any sparsity of observations with stages whose edges contain sparse observations, regardless of the actual generating process. Hence, in the absence of sufficient domain information, we recommend that situations whose emanating edges contain sparse observations should be maintained as singleton stages in the hyperstage structure until more information is available. Failing this, spurious stages may be merged by the AHC resulting in biased inference from the CEG.

In Eldridge et al. (2005), a Markov chain was used to analyse the falls intervention. Unlike the BN, Markov chains can satisfactorily express asymmetric information introduced by structural asymmetries. However, an essential property which it lacks is the ability to read conditional independence statements from the topology of its state transition

diagram. It also does not admit causal manipulations. These properties are particularly useful for modelling multi-factorial interventions where there are several different components of the intervention whose contributions and effects may not be trivially quantified or analysed. Hence the CEG as described in this chapter proves to be a more appropriate choice of model for the falls intervention compared to a Markov chain or a BN. However, most importantly, the states of a Markov process are typically elicited *before* the analysis begins. Thus, the CEG model selection may be viewed as an alternative method for identifying the states of its associated Markov chain.

The falls intervention CEG described in this chapter caters to a short-term analysis. However, this intervention represents a longitudinal process and it would be more appropriate for it to be treated as such. Dynamic variants of the CEG such as those described in Section 3.4 could be suitable candidates for modelling the falls intervention. Further, we observe that different individuals in the falls intervention may take different amounts of time to transition between the same situations. For instance, different individuals living in the community who have been assessed, referred and treated may experience a fall after varying amounts of time since they received their treatment. Also, as already noted in Section 3.4, in such a setting, it may be more natural to record events as they occur rather than at regular intervals. This type of setting corresponds more closely to a semi-Markov process rather than a Markov chain. However, like a Markov chain, a semi-Markov process does not allow for reading conditional independencies from the topology of its state transition diagram. Extended DCEGs which have a corresponding semi-Markov representation are a special subclass of the more general continuous time DCEGs described in the next chapter. Therein we further demonstrate why the general class of continuous time DCEGs would be an ideal candidate for modelling a longitudinal extension of the falls intervention.

Chapter 5

Continuous Time DCEGs

We concluded Chapter 4 with a brief motivation for a continuous time dynamic variant of CEGs. As described in Section 3.4, extended DCEGs which were introduced in Barclay et al. (2015), evolve in continuous time but have some critical limitations. One such limitation is that extended DCEGs assume that the conditional holding time distributions for two situations are the same whenever they belong to the same stage set (see Definition 3.3). Clearly, this excludes the possibility of two situations having the same conditional transition probabilities but different conditional holding time distributions. Thus, extended DCEGs define a subclass of the more general class of continuous time dynamic CEGs (CT-DCEGs). This chapter is dedicated to exploring the CT-DCEG class of models.

We begin this chapter by further motivating the need for the general class of CT-DCEGs in Section 5.1. In Section 5.2 we formally introduce the CT-DCEG class. Here, we generalise the previous definition of a stage set presented in Definition 3.3 in order to make the CT-DCEG class more flexible. This generalised definition applies to all discrete and continuous time DCEGs (and CEGs), and allows for arbitrary conditional holding time distributions to be defined for each event. In Section 5.2.1, we discuss how time-invariant covariates such as educational background, age, socioeconomic status etc. can be incorporated into a CT-DCEG – this discussion is continued in the subsequent methodological sections. In Section 5.2.2 we then compare the CT-DCEG with existing alternative models such as the CTBN (see Section 2.3.1) and two graphical model classes developed for event history analysis.

In Section 5.3 we discuss Bayesian conjugate updating of the CT-DCEG model parameters, and present a special case of model selection for this class. In Section 5.4 we demonstrate how the CT-DCEG class enjoys a, sometimes approximate, alternative representation as a semi-Markov process; extending the work presented in Barclay et al. (2015). In Section 5.5, we present a new concept of *passage-slices*, customised to our continuous

time setting, which is analogous to time-slices in discrete time settings. In this section we also demonstrate how a CT-DCEG can be "unrolled" over an arbitrary number of passage-slices, much like the unrolling of a DBN over its time-slices (Kjærulff, 1992). With these new semantics and the alternative semi-Markov representation, we then present a novel dynamic probability propagation scheme for the CT-DCEG and an associated continuous time propagation algorithm in Section 5.6 which extends the CEG propagation algorithm presented in Thwaites et al. (2008) (see Section 3.3). In Section 5.7, we use a CT-DCEG to model a longitudinal adaptation of the falls intervention containing time-invariant covariates. Finally, we conclude this chapter with a discussion in Section 5.8.

5.1 Introduction

In dynamic real-world systems, temporal effects could play a large role in the evolution of the process and contribute to our understanding of the system. To study temporal effects within a system, it is important to consider the time it takes for events to occur. The branch of statistics studying the time it takes for one or more events to happen is known as survival analysis, duration modelling or event history analysis. We can incorporate such a temporal analysis within a CEG or any of its dynamic variants by modelling the time for each transition with a *holding time* (also known as waiting time or sojourn time) random variable.

Dynamic variants of CEGs considered thus far have been for discrete time processes, except for the notable exception of extended DCEGs in Barclay et al. (2015). DCEGs were first introduced in Barclay et al. (2015). The thesis of Dr Rodrigo Collazo (Collazo, 2017) further laid down the mathematical framework of DCEGs and N time-slice DCEGs (see Section 3.4). However, dynamic continuous time variants of CEGs have not yet been explored systematically. As stated above, extended DCEGs represent a special subclass of the general CT-DCEG class as they contain only those CT-DCEGs whose situations have equivalent conditional holding time distributions whenever they are in the same stage (where a stage follows the definition presented in Chapter 3). Thus extended DCEGs systematically exclude any CT-DCEG whose situations may have equivalent transition probabilities but their corresponding conditional holding time distributions are different. For instance, two types of treatment for a particular disease may offer the same probability of recovery but one of these treatments might take a significantly longer time to show effects than the other. Another more topical example is one where a certain vaccine may need only one dose to be effective and shows effects within two weeks of the dose being administered, whereas another vaccine requires two doses which are three weeks apart and results in the same level of protection as the first vaccine but effects take a total of five weeks of the

first dose being administered. Hence, the general CT-DCEG class vastly expands the types of problems we can model compared to its extended DCEG subclass. We also note that extended DCEGs have themselves not yet been explored in great detail.

Further, while this has not been explicitly mentioned in the model descriptions of DCEGs as presented in Barclay et al. (2015) and Collazo (2017), the discrete time DCEGs they consider are such that the transitions happening out of each situation are governed by some fixed discrete holding time distribution. This appears to be an implicit assumption as the definition of stages and model likelihood in their works do not make any accommodations for the discriminatory powers endowed by having different conditional holding time distributions for each transition. Further, we can also say from their works that these distributions in fact must necessarily be geometric holding time distributions because DCEGs have been described as having the Markov property and have also been shown to have an alternative Markov process representation. When a DCEG is stratified, the geometric distributions associated with situations in different layers may have different values for their "success probability" parameters (see Appendix A). Again in this case, any information about the temporal evolution of the process modelled by such a DCEG does not have any discriminatory power and so does not need to be explicitly considered. Notice that this is also the case with vanilla CEGs. Within a discrete time DCEG or CEG, we can still allow each transition to be modelled by a different conditional non-geometric holding time distribution but this has not yet been explored.

For several discrete time longitudinal processes a fixed geometric holding time distribution for each transition out of a given situation may not be appropriate. Different transitions corresponding to distinct events, all evolving in discrete time, might still have vastly different distributions of their holding times. The general framework presented in this chapter for CT-DCEGs can also be applied to the discrete time DCEG class to incorporate non-geometric conditional holding time distributions. It is important to note, however, that if the only data available was collected at fixed regular intervals, DCEG models with geometric holding times might be appropriate even if the underlying process itself may not follow a geometric holding time for each transition. An example of this is where, on diagnosis of a chronic illness, a patient's past symptoms are recorded but the timings between the presentation of these symptoms are not recorded. See Section 6.3 for an application of a CT-DCEG where most of our observations are recorded at regular intervals while the rest are received at irregular intervals from secondary sources.

Finally, in many processes of interest, we might be interested in incorporating time-invariant covariates (existing attributes of an individual) that influence the rest of the process evolving in continuous time. Examples of such time-invariant covariates could be an individual's sex, age, chronic health conditions etc. It would seem unnatural to associate

any holding time with transitions out of the situations describing these covariates within an event tree. Throughout this chapter, we discuss how the CT-DCEG and the methodologies developed for it can accommodate such time-invariant covariates.

The methodologies in this chapter are described throughout with an adapted version of our topical infection example described below.

Example 5.1 (Infection example continued)**.** *We consider here individuals in the community and in communal establishments. Individuals in the community could get infected by one of three strains of a circulating virus. Infection caused by two of these strains can be treated, with varying success, by one of two available treatments while there are no treatments for the third strain. Individuals in communal establishments have received a vaccine protecting them against the first two strains but not against the third. We assume that hospitalisation is not required for the treatments and only the most severe cases need to be admitted to the hospital. We are interested in studying first time hospitalisations arising from the infections. Further, we assume that recovery from a prior infection from the same virus does not offer any strain-specific immunity from the same strain or cross-immunity from other strains. Figure 5.1 shows the event tree for the infection example where the three black (grey) dots indicate that after recovery the process restarts at vertex v_1 (vertex v_2) as the individual could be reinfected. Observe that vertex v_0 describes the time-invariant covariate representing the living arrangements of the individual.*

5.2 Continuous Time Dynamic Chain Event Graphs

Let $\mathcal{T}$ denote an event tree with an infinite vertex set $V(\mathcal{T})$ and an infinite directed edge set $E(\mathcal{T})$. Such event trees are also sometimes referred to as *infinite event trees*. Note that an infinite event tree might not have any terminating paths and hence, its set of leaves may be empty. Denote by $\lambda(v, v')$ a directed path from vertex v to vertex v', if it exists, where a directed path is a sequence of directed edges from v to v', and by $\Lambda(v, v')$ the set of all such paths. Denote by $\mathcal{T}_\Lambda$ the set of all root-to-leaf and all infinite paths in $\mathcal{T}$. These paths define the atoms of the event space generated by $\mathcal{T}$. Notations defined for finite event trees in Section 3.1 extend to infinite event trees in the obvious way. It is essential to first demonstrate that a probability measure can be defined on the events in the probability space of an infinite event tree model. The proof is a straightforward application of Kolmogorov's extension theorem and it is presented in Appendix B.

We now define conditional holding times for each transition in the event tree. To begin with, we assume that we have no time-invariant covariates and that each transition can be associated with a continuous holding time random variable. Each transition from vertex

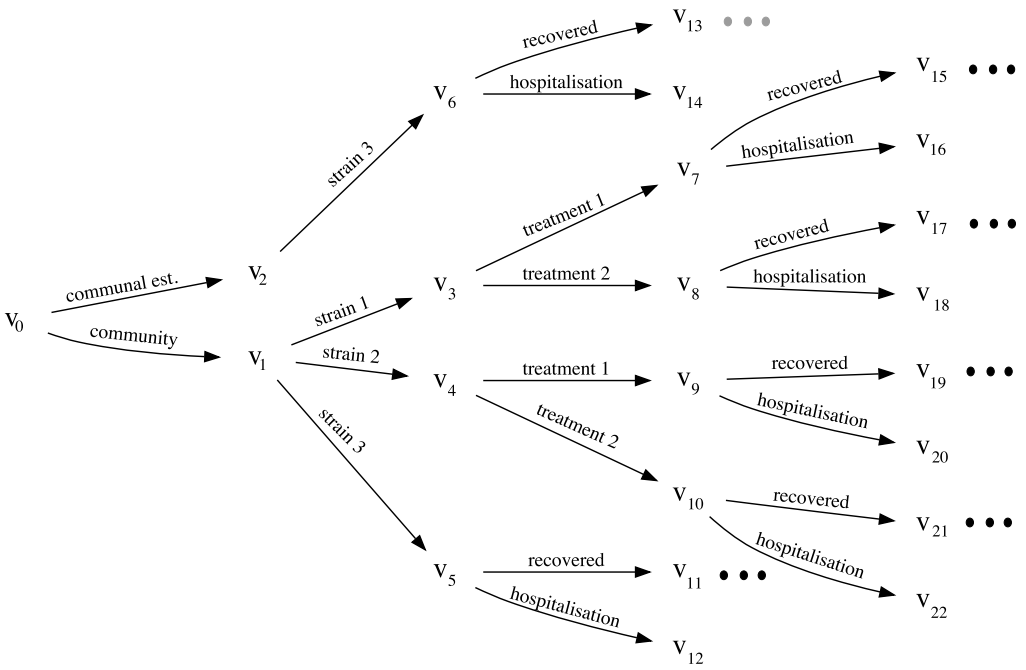


Figure 5.1: Event tree representing the longitudinal infection process.

v to vertex v' for $v \in S(\mathcal{T})$ is associated with a holding time which indicates the time spent in the state represented by situation v before transitioning along an edge $e = (v, v', l)$ to v'. This conditional holding time is represented by the continuous random variable $H(e)$. Here we assume that the holding time is dependent on the current vertex and the vertex visited next. Let $\boldsymbol{\mathcal{H}}_{\mathcal{T}} = \{\mathbf{H}(v)|v \in S(\mathcal{T})\}$ where $\mathbf{H}(v) = \{H(e)|e = (v, v', l) \in E(\mathcal{T}), v' \in \text{ch(v)}\}$ denotes the set of conditional holding time variables for each edge emanating from situation $v \in S(\mathcal{T})$. Let $H(v)$ denote the unconditional holding time at vertex v. Further, we assume that the conditional transition probabilities are independent of the conditional holding time distributions.

Example 5.2 (Infection example continued)**.** *In the infinite event tree in Figure 5.1, consider the infinite event tree rooted at vertex v_1. This tree does not represent any time-invariant covariates. Each edge in this tree can be associated with a holding time random variable. For instance, the variable $H(e_{7,15})$ indicates how long it takes for an individual in the community, infected with strain 1 of the virus, to recover after treatment 1 has been administered.*

Recall that a stage set represents a set of situations whose one-step evolutions are identical; i.e., these situations can be considered exchangeable as far as their one-step evolutions are concerned. Further as mentioned in Section 5.1, discrete time DCEGs as introduced in Barclay et al. (2015) do not explicitly model holding times for the transitions. Therefore, the definition of a stage in Barclay et al. (2015) (as in all other existing CEG literature) only considers the equivalence of the conditional transition probabilities and the corresponding edge labels as the conditions for two or more situations to be in the same stage set. As the extended DCEG is defined as an extension of the discrete time DCEG, it inherits this definition of a stage which does not consider the equivalence of the holding times for the transitions out of the situations in the same stage set. To circumvent this issue, extended DCEGs assume that whenever two situations are in the same stage (as per the Barclay et al. (2015) definition), their conditional holding time distributions are also equivalent. In Section 5.1, we considered examples of when this might not be the case. We now generalise the definition of a stage so that it considers the equivalence of both the conditional transition probability distributions and the conditional holding time distributions.

Definition 5.3 (Stage)**.** *In an event tree $\mathcal{T}$, two situations v and v' are said to be in the same stage whenever*

- $\boldsymbol{\theta}_v = \boldsymbol{\theta}_{v'}$ *such that for $\theta(e) = \theta(e')$ we require that $e = (v, \cdot, l)$ and $e' = (v', \cdot, l)$ for some edge label l;*
- *Variables $H(e)$ and $H(e')$ for $e = (v, \cdot, l)$ and $e' = (v', \cdot, l)$ follow the same distribution.*

Observe that while the second condition given in this new definition was not part of the original definition (as in Barclay et al. (2015), other CEG literature and as given in Definition 3.3), it does not change the collection of stages obtained for vanilla CEGs or discrete time DCEGs. As described in Section 5.1, due to the implicit geometric distribution (for discrete time CEGs and DCEGs), the second condition would be satisfied by default. Just as before, each stage set is assigned a unique colour. The definitions of staged trees and positions require only minor adaptations as given below.

Definition 5.4 (Staged tree)**.** *An event tree $\mathcal{T}$ whose situations are coloured according to their stage memberships is called a staged tree $\mathcal{S}$ with $\Phi_{\mathcal{S}} = \Phi_{\mathcal{T}}$ and $\boldsymbol{\mathcal{H}}_{\mathcal{S}} = \boldsymbol{\mathcal{H}}_{\mathcal{T}}$.*

Definition 5.5 (Position)**.** *In a staged tree $\mathcal{S}$, two situations v and v' are said to be in the same position whenever we have $\Phi_{\mathcal{S}_v} = \Phi_{\mathcal{S}_{v'}}$, and the conditional holding time distributions followed by the collections of random variables $\boldsymbol{\mathcal{H}}_{\mathcal{S}_v}$ and $\boldsymbol{\mathcal{H}}_{\mathcal{S}_{v'}}$ are equivalent where $\mathcal{S}_v$ and $\mathcal{S}_{v'}$ are the coloured subtrees of $\mathcal{S}$ rooted at v and v' respectively.*

We can now directly define a CT-DCEG from its underlying staged tree.

Definition 5.6 (Continuous Time Dynamic Chain Event Graph)**.** *A continuous time dynamic chain event graph (CT-DCEG)* $\mathcal{D} = (V(\mathcal{D}), E(\mathcal{D}))$ *is defined by the tuple* $(\mathcal{S}, \mathbb{W}, \boldsymbol{\Phi}_{\mathcal{S}}, \boldsymbol{\mathcal{H}}_{\mathcal{S}})$ *with the following properties:*

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

Comparing the above to the definition of extended DCEGs in Section 3.4, it is clear that extended DCEGs are a special subclass of CT-DCEGs and they contain only those CT-DCEGs which do not have loops and for which all sets of situations satisfying the first condition of a stage as described above, necessarily satisfy the second condition as well. While extended DCEGs are defined to have no loops, this is not a strict requirement for CT-DCEGs. Consider the example below where a loop within a CT-DCEG may be sensible.

Example 5.7 (Epilepsy example)**.** *Consider a study of individuals recently diagnosed with epilepsy. Suppose we are interested in studying the probability of and time to subsequent seizures after these individuals start treatment with a type of anti-epileptic drug. According to Hughes et al. (2019), while most patients find varying degrees of improvement in their condition with anti-epileptic drugs, "30% (of epileptic patients) never enter a sustained (12-month) remission from seizures, despite multiple treatment changes". It is conceivable then that among the individuals being studied, we might have a subgroup for whom the anti-epileptic drug is not effective, and their probability of and time to subsequent seizures is fairly consistent (note that this is not always the case even among those within the 30% for whom anti-epileptic drugs are not effective). The subgraph of the CT-DCEG for this subgroup may be given by Figure 5.2.*

The notation defined for event trees and CEGs extends to CT-DCEGs in the obvious way. Recall the difference between a walk and a path as described in Section 2.2. Also recall that a path is a walk but the converse is not necessarily true. While in CEGs all discussions were around paths rather than walks, for a CT-DCEG or in fact, any DCEG we must be careful to distinguish a path from a walk. For the sake of generality, we shall use the terminology of walk except when the distinction is necessary.

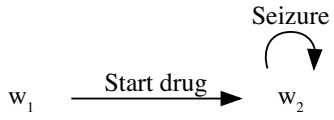


Figure 5.2: Hypothesised subgraph of the CT-DCEG for a subgroup of epileptic patients who do not benefit from the anti-epileptic drug being studied.

Observe that the definition of CT-DCEGs does not require its vertex set $V(C)$ to be finite. In the case where we have countably infinite stage sets and hence, countably infinite position sets, $V(C)$ would be infinite too. This would result in a non-finite CT-DCEG graph which poses visualisation and methodological challenges. Further, for most practical purposes we would be interested in analysing a finite sub-model of such an infinitely large model (see Section 5.5 for obtaining a finite sub-model from an infinitely large CT-DCEG). Hence, we consider here only the CT-DCEGs with a finite vertex set, hereon simply referred to as a finite CT-DCEG. Finite CT-DCEGs also have an alternative representation as a finite state semi-Markov process as described in Section 5.4. Hence, the finite CT-DCEG subclass of CT-DCEGs is of practical interest in translating a real-world domain process into a simple graphical depiction. In fact, this subclass is analogous to repeating time-slice DBNs (Dean & Kanazawa, 1989) and N time-slice DCEGs (Collazo, 2017).

Definition 5.8 (Regular CT-DCEG)**.** *Say that a CT-DCEG $\mathcal{D}$ is regular if its conditional transition parameters in $\boldsymbol{\Phi}_{\mathcal{D}}$ and its conditional holding time random variables in $\boldsymbol{\mathcal{H}}_{\mathcal{D}}$ are all time homogeneous.*

The CT-DCEGs considered in this thesis are all finite and regular.

5.2.1 Incorporating Time-Invariant Covariates

We now discuss how time-invariant covariates can be incorporated within a CT-DCEG model. Time-invariant covariates are those existing attributes of an individual that do not change with time (e.g. educational background, chronic illness, socioeconomic status) or change in a deterministic way (e.g. age). We have already noted that it is typically unnatural to associate any holding time with transitions from situations representing time-invariant covariates. This implies that the edges emanating from these situations would not have any associated holding time variables.

However, an event tree suppose a strict total ordering of events along any of its walks. A strict total ordering of say $v \prec v'$, for $\Lambda(v, v') \neq \emptyset$, implies that v is upstream

of v' in the event tree and $v' \not\preceq v$. This ordering is typically chosen such that it allows for causal interpretations to be made from the event tree and its CEG. In this case, we can interpret that $v \prec v'$ implies that *v happens before v'*. However, statistically, the event tree model does not inherently imply such a *temporal* ordering of the events. Viewed as simply a statistical model, the event tree can have any strict total ordering of its events without an associated real-world interpretation to go along with it. Indeed, this issue also arises in BNs where on the one hand it can be viewed as a computational vehicle for representing probability distributions with a visually intelligible graphical interface, and on the other, it can be viewed as a causal model where the directionality of the edges represents causal links (Korb & Nicholson, 2008). This is further complicated by the fact that BNs encoding the same probabilistic information might lead to different real-world interpretations when considered to be causal (Chickering, 1995; Korb & Nicholson, 2008). Causal discovery with CEGs for processes where these is no natural real-world ordering of the variables has been considered in Cowell and Smith (2014).

It is interesting here to note that the Christchurch Health and Development Study (CHDS) (Fergusson et al., 1986) application often used in the CEG papers (Barclay et al., 2013; Cowell & Smith, 2014; Collazo et al., 2018) models four variables all of which are time-invariant. Cowell and Smith (2014) demonstrated how an order can be chosen among these four variables for the CHDS process by maximising the chosen score function (there, the log marginal likelihood score).

We take the approach here of choosing the ordering aligned with the real-world settings of the process whenever it is possible to do so, and choosing the ordering that maximises the chosen score function (as in Cowell and Smith (2014)) when there is no associated real-world ordering available. Choosing the appropriate ordering of time-invariant covariates typically falls within the latter category. We would generally choose to place all vertices representing time-invariant covariates upstream of vertices representing events evolving in continuous time. Just as in Cowell and Smith (2014), an ordering among these vertices may then be chosen such that it maximises the model log marginal likelihood score (or any other chosen score function).

Recall from Section 5.1 that due to their Markov property, the implicit assumption made by a DCEG or a vanilla CEG modelling time-invariant covariates would be that their associated emanating edges have holding times that are governed by geometric (in discrete time processes) or exponential (in continuous time processes) distributions. Our implicit assumption here is less restrictive and simply states that the holding times associated with the edges of any time-invariant covariates are independent and identically distributed. Hence, they have no discriminatory power and need not be explicitly considered for model selection or inference (see Sections 5.3 and 5.6). Thus for two situations in the event tree represent-

ing time-invariant covariates to be in the same stage, only the first condition specified in the definition of a stage needs to be satisfied as the second is satisfied by default.

Example 5.9. *(Infection example continued) Observe that in the event tree describing the infection process in Figure 5.1, vertex v_0 represents the time-invariant covariate of an individual's living arrangements. Suppose that this event tree has the following collection of stages:*

$$\mathbb{U} = \{u_0, u_1, u_2, u_3, u_4, u_5, u_6, u_7\} \quad \textit{where}$$
$$u_0 = \{v_0\},\ u_1 = \{v_1, v_{11}, v_{15}, v_{17}, v_{19}, v_{21}, \ldots\},\ u_2 = \{v_2, v_{13}, \ldots\},$$
$$u_3 = \{v_3, v_4, \ldots\},\ u_4 = \{v_5, \ldots,\},\ u_5 = \{v_6, \ldots,\},\ u_6 = \{v_7, v_9, \ldots,\},\ u_7 = \{v_8, v_{10}, \ldots\}.$$

In the staged tree for this process, each stage u_i, $0 \leq i \leq 7$ is assigned a unique colour. In this example, all situations in the same stage are also in the same position, i.e. $\mathbb{W} = \mathbb{U}$ where for $w_i \in \mathbb{W}$ we have $w_i = u_i$ for $0 \leq i \leq 7$. For instance, situation v_1 and v_{11} have infinitely large rooted subtrees which are isomorphic in the colour and structure preserving sense, and hence they are in the same position.

Figure 5.3 shows the graph of the CT-DCEG for the infection example. Vertices representing position $w_i \in \mathbb{W}$ are labelled as w_i in the graph, for $0 \leq i \leq 7$ and the leaves are collected into the sink w_∞. While we do not do so here, note that when the collection of positions and stages are equivalent, the vertex colouring in the graph of the CT-DCEG can be suppressed without any loss of information.

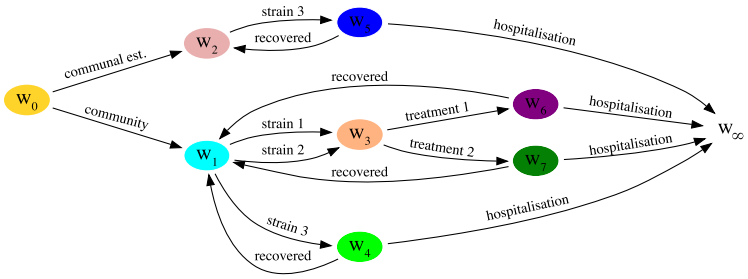


Figure 5.3: Graph of the CT-DCEG for the infection process.

5.2.2 Comparison with Existing Models

A CT-DCEG is most closely associated with continuous time BNs (CTBNs) (see Section 2.3.1) and two classes of graphical models used for event history analysis.

CTBNs were first proposed in Nodelman et al. (2002) as a continuous time alternative to DBNs. They overcome the issue of having to choose a fixed time granularity needed to construct DBNs. For many real-world processes, different components of the process evolve on different time scales and hence, describing the entire process using a fixed time granularity may not be appropriate. CTBNs, on the other hand, describe structured stochastic processes with a finite state space as they evolve over continuous time. The model is initiated with a vanilla BN and the dynamics of the evolution are described through a directed and possibly cyclic graph. The CTBN models each variable (represented by a node in the graph) as a finite state, continuous time Markov process whose transition intensities are determined by the current value of its parents in the graph as well as its own current value. These transition intensities are contained within a conditional Markov intensity matrix. However, a major shortcoming of CTBNs, as they were originally described, is that the holding time in a given state – where a state is simply a realisation of one of its variables – is always described by an exponential distribution. This issue arises by design as a Markov process naturally has an exponential holding time random variable for each state, although the parameters are allowed to vary between states. Additionally, similar to BNs, CTBNs rely on variable-based descriptions. They embed structural zeros within their conditional intensity matrices, and are unable to encode structural missing values. Recall that in Section 4.1 we have already discussed the limitations of variable-based descriptions of asymmetric processes.

There have been several notable efforts to relax the exponential restriction on the holding time distributions of CTBNs. Nodelman and Horvitz (2003), Gopalratnam et al. (2005), and Nodelman et al. (2005) approximate arbitrary holding time distributions in the CTBN with phase-type distributions by adding exponentially distributed hidden states called "phases" to their conditional intensity matrices. Other methods involve adding hidden variables to the CTBN (Nodelman et al., 2005; Liu et al., 2018) and adding holding time variables known as "clocks" (Engelmann et al., 2020). The latter is most similar to CT-DCEGs. Just as in a CT-DCEG, it adopts the approach of explicitly modelling holding times as random variables which can have arbitrary distributions rather than them being implicitly exponential through the conditional intensity matrix construction. Under this approach, a CTBN with clocks is built using continuous time semi-Markov (rather than Markov) processes with each variable having an unconditional holding time distribution and thus, allows for closed form parameter estimation and model selection when using conjugate distributions.

Despite the similarities, CT-DCEGs and CTBNs are different in the type of processes for which they are best suited. Apart from not being able to explicitly accommodate structural zeros and structural missing values, there does not exist a method of exploring

conditional independencies within a CTBN. This is because the CTBN has a variable-based description, and so exploring conditional independencies within its structure is complicated by the fact that over its temporal evolution, the states of all the variables become correlated. This problem is typically referred to as *temporal entanglement* (Boyen & Koller, 1998; Nodelman et al., 2002). Hence, a CTBN suffers from the same shortcomings as a BN in expressing context-specific conditional independencies within its graph topology. Further, in a CTBN the state of a variable depends on the current state of its parents and that of itself, and the holding times variables describe the time taken to transition between states of the same variable. This is vastly different to the way a CT-DCEG models a process. Finally, while CTBNs do not allow multiple transitions to occur at the same time, they do not otherwise require any ordering in the transitions experienced by its variables. Hence, a CTBN is preferable for processes without significant structural asymmetries where the interest is in modelling the state of a dynamic system as a whole, and where the interest is in modelling holding times from a variable assuming one value to another. Whereas, a CT-DCEG is more suitable to dynamic processes which naturally follow a total (strict or otherwise) ordering of the events which transpire during the evolution of the process, and where the flexibility of allowing arbitrary holding time distributions for each event (rather than each variable) might be useful.

We next move our attention to graphical models for the analyses of event history data. Event history data consist of sequences of certain events of interest along with their time of occurrence. Graphical models used to represent the events contained in such data were first introduced as *graphical duration models* (Gottard, 2007) and as *local independence graphs* (Didelez, 2008). Note that the concept of local independence was first introduced by Schweder (1970) and later generalised by Aalen (1987). Consider finite state space stochastic processes $X = \{X_t\}$, $Y = \{Y_t\}$ and $Z = \{Z_t\}$ which evolve over continuous time such that multiple transitions cannot occur simultaneously. We can say that X is locally independent, over some time interval, of Y given Z if given the information about whether and when events have occurred in Z in the time interval, the transition intensities for changes in X are independent of the value of Y for all time in the interval. Hence, local independence is one-sided and not symmetric whereas stochastic independence (described in Section 2.2.2 as independence) is symmetric. Both the graphical duration and local independence graphs model the data by marked point processes and the graphs representing these models depict the marks or events as vertices. Graphical duration models represent the conditional independence structure of the data using a special class of chain graph models and they represent both local and stochastic independence structures using a combination of directed and undirected edges. On the other hand, local independence graphs, as the name suggests, represent local independencies where directed edges depict

local dependence. While these graphs model processes that are very similar to those modelled by CT-DCEGs and are able to represent context-specific information, their framework does not allow for expressing how conditional independence relationships can change over time. In fact, modelling of local dependence makes the independence structures represented by these graphs more similar to DBNs. Further note that in these graphs, the events are represented by vertices whereas in the CT-DCEG they are represented by edges.

We now briefly describe alternative graphical models for processes evolving in continuous time. Event-driven CTBNs (Bhattacharjya, Shanmugam, et al., 2020) model the effect of irregularly occurring external events on the evolution of the system variables of a CTBN (e.g. frequency of meals and physical activities affecting a diabetic patient's blood glucose level) by modelling the events as nodes within the CTBN with specified conditional intensity rates. They suffer similar shortcomings to CTBNs. Piecewise-constant conditional intensity models (Gunawardana et al., 2011) are a class of marked point processes that capture the temporal dependence of an event – represented by a decision tree – on past events through a set of piecewise-constant intensity functions. From Qin and Shelton (2015), we can infer that these models can express context-specific independencies through their decision tree representations. However, similar to staged tree (see Section 3.1.2), they can easily get unwieldy for large processes with significant asymmetries. Graphical event models (Meek, 2014) constitute of a dynamic directed graph over a set of events where the edges represent potential dependencies, and a statistical model whose parameters are the intensity functions for each event conditioned on its parent events. These models are similar to the other graphical models for event history data discussed above and suffer from similar limitations. State variable graphical event models (Bhattacharjya, Subramanian, et al., 2020) are a flexible modelling framework that extends event-driven CTBNs/CTBNs to a non-Markovian setting and generalise graphical event models. The exact nature of the non-Markovian dependence is to be defined by the modeller, and this choice affects the scope and limitations of the model. Lastly, we discuss temporal nodes BNs (TNBNs) (Arroyo-Figueroa & Sucar, 1999) are a visionary but currently underdeveloped class of graphical models. A TNBN contains *temporal nodes* which are ordered pairs of the realisation of a variable and the time interval for which it assumes this value, similar to the concept of holding times. Although some properties of propagation in a TNBN are similar in intuition to that in a CT-DCEG (see Section 5.6), they are presented in a non-technical way with no formal justification. Primarily, similar to a CT-DCEG, evidence in the TNBN leads to the creation of a simpler TNBN, and temporal evidence leads to a revision of past transition probabilities. However, the TNBN has several shortcomings. The occurrence of an event at a particular instant does not constitute direct evidence in a TNBN (Galán & Díez, 2002). Further, TNBNs lack the formalisation of standard models used for temporal processes.

5.3 Conjugate Learning and Model Selection

5.3.1 Conjugate Learning

We shall now consider conjugate updating of the parameters of a CT-DCEG model. The work presented in this subsection is analogous to the conjugate updating described in Barclay et al. (2015) for extended DCEGs. Here we show that it extends to the entire CT-DCEG class in a straightforward way. The conjugate updating will be necessary for the model selection methodology presented in Section 5.3.3.

Consider a CT-DCEG $\mathcal{D}$ which has no vertices representing time-invariant covariates. Let $\mathbb{U} = \{u_1, u_2, \ldots, u_k\}$ be its collection of stages such that each stage u_i has k_i emanating edges. Suppose we have a complete random sample of n individuals. For each individual $1 \leq m \leq n$, their data is given by the following sequence of tuples

$$\rho_m = ((e^m_{j_1k_1}, h^m_{j_1k_1}), (e^m_{j_2k_2}, h^m_{j_2k_2}), \ldots (e^m_{j_{l_m}k_{l_m}}, h^m_{j_{l_m}k_{l_m}})),$$

where the first element of each tuple represents the edge traversed by the individual such that e_{jk} represents the individual traversing kth edge emanating from vertex v_j, and the second element gives the holding time associated with that edge. Here we assume that for each individual m where $1 \leq m \leq n$, the vertex v_{j_1} always corresponds to the root vertex of the graph of the CT-DCEG. The data from the n individuals can be summarised as $\mathbf{y} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_k\}$ where $\mathbf{y}_i = (\mathbf{n}_i, \mathbf{h}_i)$ corresponds to the observations for stage u_i, $i = 1, 2, \ldots, k$. Here, $\mathbf{n}_i = (\mathbf{n}_{i1}, \mathbf{n}_{i2}, \ldots, \mathbf{n}_{ik_i})$ and $\mathbf{h}_i = (\mathbf{h}_{i1}, \mathbf{h}_{i2}, \ldots, \mathbf{h}_{ik_i})$ where $\mathbf{n}_{ij}$ and $\mathbf{h}_{ij}$ correspond to the observations for the jth edge emanating from stage u_i for $1 \leq j \leq k_i$. Each $\mathbf{n}_{ij}$ is a vector of ones with length equal to the number of individuals $\bar{\mathbf{n}}_{ij}$ who traverse the jth edge emanating from stage u_i, and $\mathbf{h}_{ij}$ is a vector of the holding times for each of the $\bar{\mathbf{n}}_{ij}$ individuals.

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

$$p(\mathbf{y} \,|\, \boldsymbol{\Phi}_{\mathcal{D}}, \boldsymbol{\Pi}_{\mathcal{D}}, \mathcal{D}) = \prod_{i=1}^{k} p(\mathbf{y}_i \,|\, \boldsymbol{\theta}_i, \boldsymbol{\pi}_i, \mathcal{D}). \tag{5.1}$$

As in the case of vanilla CEGs in Section 3.2.1, we assume here that the conditional transition and holding time parameters are *a priori* mutually independent. Of course, in certain

scenarios this assumption might not hold but we consider the simplest scenario here. It follows under the separability of the likelihood above that they will also be mutually independent *a posteriori*. With this we can write

$$\begin{aligned} p(\mathbf{y}_i \,|\, \boldsymbol{\theta}_i, \boldsymbol{\pi}_i, \mathcal{D}) &= \prod_{j=1}^{k_i} p(\mathbf{n}_{ij}, \mathbf{h}_{ij} \,|\, \theta_{ij}, \pi_{ij}, \mathcal{D}) \\ &= \prod_{j=1}^{k_i} p(\mathbf{h}_{ij} \,|\, \pi_{ij}, \mathcal{D}) p(\mathbf{n}_{ij} \,|\, \theta_{ij}, \mathcal{D}) \\ &= \prod_{j=1}^{k_i} \prod_{l=1}^{\bar{\mathbf{n}}_{ij}} \Big\{ p(h_{ijl} \,|\, \pi_{ij}, \mathcal{D}) \times p(n_{ijl} \,|\, \theta_{ij}, \mathcal{D}) \Big\}. \end{aligned} \tag{5.2}$$

Thus the likelihood of the model separates into the likelihoods of the conditional transition and conditional holding time parameters. This conveniently allows us to estimate the conditional transition and conditional holding time parameters independently.

The conditional transition parameters can now be updated exactly as in Section 3.2.1. We shall not repeat that here in the same detail. Suffice to say that $\boldsymbol{\theta}_i$ has a Dirichlet prior distribution with hyperparameter $\boldsymbol{\alpha}_i = (\alpha_{i1}, \alpha_{i2}, \ldots, \alpha_{ik_i})$ and a Dirichlet posterior distribution with hyperparameter $\boldsymbol{\alpha}_i^* = (\alpha_{i1}^*, \alpha_{i2}^*, \ldots, \alpha_{ik_i}^*)$ where $\alpha_{ij}^* = \alpha_{ij} + \bar{\mathbf{n}}_{ij}$ for $1 \leq i \leq k$ and $1 \leq j \leq k_i$.

Assume that the continuous conditional holding times for stage $u_i \in \mathbb{U}$ come from a Weibull distribution with known shape parameter κ_{ij} and unknown scale parameter π_{ij}, for $1 \leq i \leq k$ and $1 \leq j \leq k_i$. This gives us

$$\begin{aligned} p(\mathbf{h}_{ij} \,|\, \pi_{ij}, \mathcal{D}) &= \prod_{l=1}^{\bar{\mathbf{n}}_{ij}} p(h_{ijl} \,|\, \pi_{ij}, \mathcal{D}) \\ &= \prod_{l=1}^{\bar{\mathbf{n}}_{ij}} \frac{\kappa_{ij}}{\pi_{ij}} (h_{ijl})^{\kappa_{ij}-1} \exp\left(\frac{-{h_{ijl}}^{\kappa_{ij}}}{\pi_{ij}} \right). \end{aligned} \tag{5.3}$$

Under a conjugate setting, the scale parameter π_{ij} of the Weibull distribution, has an Inverse-Gamma prior distribution with shape hyperparameter β_{ij} and scale hyperparameter γ_{ij}. The density of this prior distribution is given as follows

$$p(\pi_{ij}|\mathcal{D}) = \frac{{\gamma_{ij}}^{\beta_{ij}}}{\Gamma(\beta_{ij})} (\pi_{ij})^{-\beta_{ij}-1} \exp\left(\frac{-\gamma_{ij}}{\pi_{ij}} \right). \tag{5.4}$$

Thus we can find the posterior of π_{ij} as

$$\begin{aligned} p(\pi_{ij}|\mathbf{h}_{ij}, \mathcal{D}) &\propto p(\pi_{ij}|\mathcal{D}) \prod_{l=1}^{\bar{\mathbf{n}}_{ij}} p(h_{ijl}|\pi_{ij}, \mathcal{D}) \\ &\propto {\pi_{ij}}^{-\beta_{ij}-1} \exp\left(\frac{-\gamma_{ij}}{\pi_{ij}}\right) \prod_{l=1}^{\bar{\mathbf{n}}_{ij}} \frac{1}{\pi_{ij}} \exp\left(\frac{-{h_{ijl}}^{\kappa_{ij}}}{\pi_{ij}}\right) \\ &= {\pi_{ij}}^{-\beta_{ij}-\bar{\mathbf{n}}_{ij}-1} \exp\left(\frac{-(\gamma_{ij} + \sum_l {h_{ijl}}^{\kappa_{ij}})}{\pi_{ij}}\right) \end{aligned} \tag{5.5}$$

which is an Inverse-Gamma distribution with shape hyperparameter $\beta^*_{ij} = \beta_{ij} + \bar{\mathbf{n}}_{ij}$ and scale hyperparameter $\gamma^*_{ij} = \gamma_{ij} + \sum_{l=1}^{\bar{\mathbf{n}}_{ij}} (h_{ijl})^{\kappa_{ij}}$.

The marginal likelihood of the model is available in closed form as

$$\begin{aligned} p(\mathbf{y}|\mathcal{D}) &= \int_{\mathbf{\Pi}_\mathcal{D}} \prod_{i=1}^{k} \prod_{j=1}^{k_i} \Big\{ p(\mathbf{h}_{ij}|\pi_{ij}, \mathcal{D}) p(\pi_{ij}|\mathcal{D}) \Big\} d\mathbf{\Pi}_\mathcal{D} \times \int_{\mathbf{\Phi}_\mathcal{D}} \prod_{i=1}^{k} \Big\{ p(\mathbf{n}_i|\boldsymbol{\theta}_i, \mathcal{D}) p(\boldsymbol{\theta}_i|\mathcal{D}) \Big\} d\mathbf{\Phi}_\mathcal{D} \\ &= \prod_{i=1}^{k} \left\{ \frac{\Gamma(\beta^*_{ij})(\gamma_{ij})^{\beta_{ij}} (\kappa_{ij})^{\bar{\mathbf{n}}_{ij}} \prod_l (h_{ijl})^{\kappa_{ij}-1}}{\Gamma(\beta_{ij})(\gamma^*_{ij})^{\beta^*_{ij}}} \times \frac{\Gamma(\bar{\boldsymbol{\alpha}}_i)}{\Gamma(\bar{\boldsymbol{\alpha}}^*_i)} \prod_{j=1}^{k_i} \frac{\Gamma(\alpha^*_{ij})}{\Gamma(\alpha_{ij})} \right\}. \end{aligned} \tag{5.6}$$

We now extend the above discussion to consider the inclusion of time-invariant covariates in the CT-DCEG. Suppose that the CT-DCEG $\mathcal{D}$ also has time-invariant covariates. Further, let u_i for $1 \leq i \leq k' < k$ represent the stage sets of situations with holding times and u_i for $k' < i \leq k$ represent the stage sets of situations without holding times (precisely those representing time-invariant covariates). Then we can simply modify the marginal likelihood of the model given in Equation 5.6 as

$$p(\mathbf{y}|\mathcal{D}) = \prod_{i=1}^{k'} \left\{ \frac{\Gamma(\beta^*_{ij})(\gamma_{ij})^{\beta_{ij}} (\kappa_{ij})^{\bar{\mathbf{n}}_{ij}} \prod_l (h_{ijl})^{\kappa_{ij}-1}}{\Gamma(\beta_{ij})(\gamma^*_{ij})^{\beta^*_{ij}}} \right\} \times \prod_{i=1}^{k} \left\{ \frac{\Gamma(\bar{\boldsymbol{\alpha}}_i)}{\Gamma(\bar{\boldsymbol{\alpha}}^*_i)} \prod_{j=1}^{k_i} \frac{\Gamma(\alpha^*_{ij})}{\Gamma(\alpha_{ij})} \right\}. \tag{5.7}$$

Note that we have chosen a Weibull conditional holding time distribution above for the convenience of a conjugate analysis. Within this, we assume that the shape parameter is known. The shape parameter κ of a Weibull distribution can be interpreted as follows: $\kappa = 1$ simplifies to the exponential distribution where the rate of transition is assumed to be constant, $\kappa < 1$ indicates that the rate of transition decreases with time whereas $\kappa > 1$ indicates that the rate increases with time. The transition rate determined by the shape parameter of the Weibull conditional holding time distribution along an edge is not to be confused with the conditional transition probability of the edge which gives the limiting probability of an individual actually going through that specific transition along that edge. Other conjugate distributions (such as the exponential and Gamma, or log-Normal with

known precision and Normal) may also be used. A non-conjugate analysis with an arbitrary holding time distribution which do not have a conjugate prior is also possible using MCMC methods and is discussed further in Chapter 7. This work is ongoing and is not reported in detail within this thesis. A Bayesian non-parametric treatment of the holding times is also possible, see for example Kalbfleisch (1978) and Muliere and Walker (1997).

5.3.2 Prior Specification

The hyperparameters $(\alpha_{i1}, \alpha_{i2}, \ldots, \alpha_{ik_i})$ for the Dirichlet priors for $\boldsymbol{\theta}_i$ associated with stage u_i can be set using the mass conservation property and the imaginary sample size as described in Section 3.2.2.

We next look into specifying the hyperparameters for the unknown scale parameter of the Weibull holding time distribution. Recall first that π_{ij} is the scale parameter of the Weibull distribution (with known shape parameter κ_{ij}) followed by the conditional holding time variable on the jth edge emanating from some representative situation in stage u_i. The hyperparameters β_{ij} and γ_{ij} of the Inverse-Gamma prior distribution for π_{ij} are directly linked to the expected mean and variance of the holding times along its associated edge (Barclay et al., 2015). If there is prior domain knowledge, this can be incorporated into the hyperparameters by eliciting the expected mean and variance from the domain experts.

When such information is not available, we may determine how to set these hyperparameters based on how they are updated in the conjugate prior-to-posterior analysis. Since, the shape hyperparameter β_{ij} is updated in the same way as the corresponding Dirichlet hyperparameter α_{ij}, we can set β_{ij} to be equivalent to α_{ij}. This is analogous to the approach taken by Barclay et al. (2015). The scale hyperparameter γ_{ij} is updated as $\gamma^*_{ij} = \gamma_{ij} + \sum_{l=1}^{\bar{\mathbf{n}}_{ij}} (h_{ijl})^{\kappa_{ij}}$. Similar to the interpretation of the Dirichlet hyperparameters as "pseudo-counts" or the *strength of the prior belief*, we could set the scale hyperparameter as ${\iota_{ij}}^{\kappa_{ij}}$ where we interpret ι_{ij} as a "pseudo-holding time". This approach applies a similar treatment to our pseudo-holding time as we do for any observed holding time under the conjugate analysis. We note here that Barclay et al. (2015) present an alternative approach for setting the scale hyperparameter of the Inverse-Gamma distribution. They set a common prior mean of 1 for each Inverse-Gamma distribution. As the mean of the Inverse-Gamma is given by $\frac{\gamma_{ij}}{(\beta_{ij}-1)}$, the scale parameter γ_{ij} is effectively set as $\beta_{ij} - 1$.

Recall that for model selection using the AHC algorithm (see Section 3.2.3), each situation in the event tree is initially considered to be a singleton stage and at each step, stages offering the best improvement to the log marginal likelihood score are merged together. For an infinite event tree, prior setting as described above becomes an impossible task. Barclay et al. (2015) addressed this issue by describing how the hyperparameters can be set for a given candidate extended DCEG model. The Dirichlet hyperparameters can be

set based on the limiting distribution of the Markov process associated with the extended DCEG without considering its holding time distributions. This method is useful if we have a small finite number of candidate models that we wish to compare. If our model selection is to start from the event tree itself as is typically done in the vanilla CEG case, we need heuristic approaches to make searching through the vast model search space of a CT-DCEG feasible. We describe one such special case of model selection in the next subsection.

5.3.3 Model Selection

Model selection in the CEG family is equivalent to identifying the collection of stages as a CEG is completely defined by its staged tree as described in Section 4.3. The CT-DCEG model search space is extremely vast as it has an underlying event tree that is infinitely large. To search through this space effectively, we need to restrict it to the models which are of practical interest.

In this thesis, we consider a special case of finite CT-DCEGs in which model selection is simplified. We first need to define two new concepts of an *invariant subtree* and a *repeating subtree*. These enable an object-oriented description of the continuous time infinite event tree. Object-oriented approaches have been very successful for modelling large complex processes and to speed up inference. See Koller and Pfeffer (1997) and Bangsø and Wuillemin (2000) for object-oriented BNs, and Collazo (2017) for a detailed object-oriented approach to discrete time DCEGs.

Definition 5.10 (Invariant Subtree)**.** *Say that a subtree $\mathcal{T}^I$ of an event tree $\mathcal{T}$ rooted at v_0 – the root of $\mathcal{T}$ – is invariant if it is the largest subtree such that all events described by $\mathcal{T}$ appear at most once along each root-to-leaf walk in $\mathcal{T}^I_\Lambda$.*

If the edge labels are assigned such that no two distinct events have the same edge label (e.g. labelling as "high (low) risk" and "high (low) socio-economic status" rather than "high (low)" for both), the condition in the above definition simplifies to an edge label appearing at most once along each root-to-leaf walk in $\mathcal{T}^I_\Lambda$. The invariant subtree thus describes all the events that might be experienced by an individual, along any walk starting from the root, before they observe some event for the second time. For a finite CT-DCEG, the invariant subtree of its underlying event tree will also be finite.

Definition 5.11 (Repeating Subtree)**.** *Say that a subtree $\mathcal{T}^R$ of an event tree $\mathcal{T}$ is a repeating subtree if it is rooted at a leaf of the invariant subtree and it is the largest subtree such that all events described by $\mathcal{T}$ appear at most once along each root-to-leaf walk in $\mathcal{T}^R_\Lambda$.*

Thus a repeating subtree is in fact part of the invariant subtree. This makes the term "invariant" slightly imprecise as each repeating tree is itself a subtree of the invariant tree.

While an event tree has only one invariant subtree, it can have multiple distinct repeating subtrees rooted at different leaves of the invariant subtree. Denote by $\mathbb{R}(\mathcal{T})$ the collection of distinct repeating subtrees of the event tree $\mathcal{T}$.

Example 5.12 (Infection example continued)**.** *For the infection process, the invariant subtree is given by the subtree induced by the vertices in the set* $\{v_i\}$, $0 \leq i \leq 22$. *This is isomorphic to the event tree in Figure 5.1 without considering the three dots indicating the repetitions in the dynamic process. The repeating subtrees are shown in Figure 5.4.*

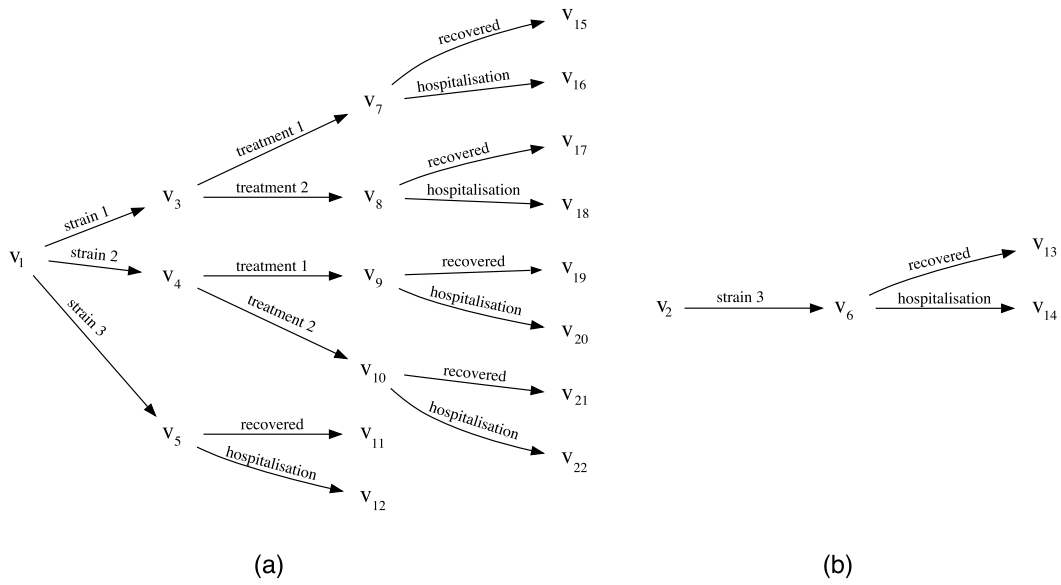


Figure 5.4: The subtrees show the two repeating subtrees for the infection process.

Notice how the concepts of invariant and repeating subtrees are closely associated with the idea of time-slices in discrete time dynamic models. The invariant subtree corresponds to the initial "time-slice" and the repeating subtrees are utilised to describe all the subsequent "time-slices". In this way, it shares some similarities with a 2-time-slice DBN (see Section 2.3.1) which is described by an initial BN and a transition model. This discussion is continued in Section 5.5 where we present a continuous time analogue of time-slices.

In the simplest case we can assume that the parameters of each repeating subtree are equivalent to the parameters of the subtree of the invariant tree to which it is structurally isomorphic. Such an assumption greatly simplifies the model selection process. It implies that the repeating parts of the underlying infinite event tree are not only structurally but also

parametrically equivalent. Under this assumption, observations associated with the repeating subtrees can be pooled with the corresponding observations for the invariant subtree for model selection and parameter estimation. Hyperparameter specification for the Dirichlet priors on the invariant subtree proceeds as in Section 3.2.2 and for the Inverse-Gamma priors as in Section 5.3.2. Identifying the collection of stages for the invariant subtree with the pooled data follows the procedure described in Section 3.2.3 for the vanilla CEG, using the marginal likelihood (or equivalently, its logarithm) given in Equation 5.7 as the score to be maximised. To obtain the graph of the CT-DCEG we first follow a construction from the staged invariant tree which is identical to the construction of a CEG from a staged tree (see Section 4.3). However, this would result in the cyclic edges also being collected into the sink vertex. To obtain the graph of the required CT-DCEG from this CEG, we would need to change the vertex entered by each cyclic edge with reference to the repeating subtrees.

There are several processes for which such an assumption might not be appropriate. For instance, the graphical structure of the model depicting the process might be fixed but the parameters of the distributions might change depending on the time index itself in a regular pattern (e.g. age related changes each year) or irregularly (e.g. irregularly occurring external events such as shortage of medications or reduction in funding for a specific type of treatment). In other cases, the structure of the graph itself may also be subject to change over time. Such changes are relatively easier to implement in a discrete time setting (see e.g. J. W. Robinson and Hartemink (2008) and Grzegorczyk and Husmeier (2009) for non-stationary DBNs, and Freeman and Smith (2011b) for dynamically evolving staged trees) than in continuous time. Hence, in this thesis, we begin by exploring model selection in CT-DCEGs within this simplified setting. A detailed discussion on this is presented in Section 5.8.

Example 5.13 (Infection example continued)**.** *The above assumption would imply in our infection process for instance that for an individual in the community the probability of recovering from strain i ($i = 1, 2$) after receiving treatment j ($j = 1, 2$) is not dependent on being infected by any strain of this virus before. Under our previous assumption of no immunity acquired by infection, this new assumption would appear to be sensible. This assumption enables us to consider each bout of infection for an individual independently and thus adds to the data available to compare models and to estimate parameters. For instance, if we observed an individual in the community who got infected three times and another individual, also in the community, who got infected twice, with the independence assumption it is equivalent to having observed a single bout of infection in five individuals in the community.*

5.4 A Semi-Markov Representation

Before discussing inference for the CT-DCEG class, it is of interest to note that CT-DCEGs have an alternative representation as a semi-Markov process (SMP). This section extends the work presented in Barclay et al. (2015) where extended DCEGs whose graphs are simple graphs were shown to have an SMP representation. Since the state transition diagram of an SMP must necessarily be a simple graph (loops, however, are allowed), the translation of an extended DCEG with a simple graph into an SMP is relatively straightforward. Here we show how all CT-DCEGs (including the extended DCEG subclass) have an alternative SMP representation even when their graphs are multigraphs. Such an SMP representation, however, is an approximation of the CT-DCEG model when its graph is a multigraph. The associated SMP representation of a CT-DCEG allows us to leverage well-developed methodologies from the SMP literature to answer queries for which either corresponding CT-DCEG techniques are yet to be developed or where a macro-level estimate obtained from the, often approximate, SMP representation is sufficient.

An SMP has two simultaneously evolving sub-processes. One concerns the state occupied by an individual, the other the time spent in each state. Consider a stochastic process $\mathbf{Z} = \{Z_t, t \geq 0\}$ on a discrete state space $\mathbf{S}$. The state occupied at the nth transition is given by $\mathbf{X} = (X_n)_{n\in\mathbb{N}}$, the jump times by $\mathbf{T} = (T_n)_{n\in\mathbb{N}}$ and the holding time in X_n before moving to X_{n+1} by $\boldsymbol{\tau} = (\tau_n)_{n\in\mathbb{N}}$ where $\tau_n = T_n - T_{n-1}$. The process $\mathbf{Z}$ is an SMP when

$$p(X_{n+1} = j, \tau_{n+1} \leq t \,|\, X_n, \ldots, X_0; \tau_n, \ldots, \tau_1) = p(X_{n+1} = j, \tau_{n+1} \leq t \,|\, X_n = i), \tag{5.8}$$

where $n \geq 1, t \geq 0$ and $i, j \in \mathbf{S}$. The process $\mathbf{X}$ is called the *embedded Markov chain* with transition probability matrix $P = (p_{ij})$ where $p_{ij} = p(X_{n+1} = j \,|\, X_n = i)$. An SMP is completely defined by its renewal kernel $Q(t) = [Q_{ij}(t) \,|\, i, j \in \mathbf{S}]$ and its initial distribution $p = [p_i \,|\, i \in \mathbf{S}]$ where $p_i = p(X_0 = i)$. The (i, j)th entry of the renewal kernel Q is

$$Q_{ij}(t) = p(X_{n+1} = j, \tau_{n+1} \leq t \,|\, X_n = i) = p_{ij} F_{ij}(t), \tag{5.9}$$

where $F_{ij}(t) = p(\tau_{n+1} \leq t \,|\, X_{n+1} = j, X_n = i)$ is the cumulative distribution function of the random variable representing the holding time in state i before transitioning to state j.

Thus in an SMP, the successive states occupied are determined by the transition probabilities of its embedded Markov chain whereas the holding times are dependent on the current state and the state occupied next. This allows the flexibility to use arbitrary non-exponential (in continuous time) or non-geometric (in discrete time) holding time distributions. The Markov property is only required at the transition times between the states. Hence, an SMP is not strictly Markovian (Moura & Droguett, 2008).

First we consider a CT-DCEG $\mathcal{D} = (V(\mathcal{D}), E(\mathcal{D}))$ with no time-invariant covariates. Let the SMP representation corresponding to the CT-DCEG $\mathcal{D}$ be given by the stochastic process $\mathbf{Z}$ with state space given by $\mathbf{S} = V(\mathcal{D})$. A transition from state v_i to state v_j for $v_i, v_j \in \mathbf{S}$ is possible if and only if there exists at least one edge between v_i and v_j in $E(\mathcal{D})$ for $v_i, v_j \in V(\mathcal{D})$. Recall that while the graph of a CT-DCEG can be a multigraph, the state transition diagram of an SMP must be a simple graph although loops are permitted. Suppose there exist m edges from vertex v_i to vertex v_j in $E(\mathcal{D})$, denoted as edges e_{ijl} where $1 \leq l \leq m$. Let edge e_{ijl} have conditional transition parameter θ_{ijl} and holding time random variable H_{ijl}. We can then set the transition probability from v_i to v_j in the SMP as follows

$$p_{ij} = \sum_{l=1}^{m} \theta_{ijl}, \tag{5.10}$$

and the holding time distribution, denoted by the variable G, at state v_i conditional on a transition to state v_j as the following finite mixture distribution

$$f_G(t) = \sum_{l=1}^{m} \theta_{ijl} \, f_{H_{ijl}}(t), \tag{5.11}$$

where $f_{H_{ijl}}$ is the probability density associated with holding time variable H_{ijl}.

Notice that combining multiple edges with the same directionality between two vertices, say v_i and v_j for the SMP representation results in losing some information embedded in the CT-DCEG model. Hence, in the case where the graph of the CT-DCEG is not simple, such an alternative representation is only an approximation. We next prove that the SMP obtained from the process detailed above is valid.

Theorem 5.14. *The SMP representation $\mathbf{Z}$ of a CT-DCEG $\mathcal{D} = (V(\mathcal{D}), E(\mathcal{D}))$ with no time-invariant covariates, as described above, satisfies the definition of a semi-Markov process.*

Proof. By construction, the representation $\mathbf{Z}$ of the CT-DCEG $\mathcal{D}$ has state space $\mathbf{S} = V(\mathcal{D})$. Recall that an SMP is completely defined by its initial distribution and its renewal kernel.

Let $v_i, v_j \in \mathbf{S}$. The transition matrix $\mathrm{P} = (p_{ij})$ for the SMP $\mathbf{Z}$ is given by entries

$$p_{ij} = \begin{cases} \theta^*_{ij}, & \text{if } \mathbb{1}_{\mathcal{D}}(v_i \to v_j) = 1 \\ 0, & \text{otherwise} \end{cases}$$

where $\mathbb{1}_{\mathcal{D}}(v_i \to v_j)$ indicates whether there exists at least one edge from v_i to v_j in the CT-DCEG $\mathcal{D}$ and $\theta^*_{ij} = \sum_{l=1}^{m} \theta_{ijl}$ where θ_{ijl} is the transition probability associated with edge e_{ijl}, $l = 1, \ldots, m$ for the m edges from v_i to v_j in $\mathcal{D}$ and $m \geq 1$. The transition probabilities for the emanating edges from each vertex $v_i \in \mathbf{S}$ necessarily sum to 1 as the conditional

transition parameters $\theta_{i,\cdot,\cdot}$ sum to 1 in the CT-DCEG model.

The cumulative conditional holding time distribution at edge e_{ij} from state v_i to state v_j in $\mathbf{Z}$ is

$$F_{ij}(t) = \mathbb{P}(\tau_{ij} \leq t \,|\, X_{n+1} = v_j, X_n = v_i), \tag{5.12}$$

where the holding time τ_{ij} in $\mathbf{Z}$ is governed by the holding time mixture distribution G with density given in Equation 5.11.

The entries of the renewal kernel $Q(t)$ of the SMP representation $\mathbf{Z}$ are thus given by

$$Q_{ij}(t) = p_{ij} F_{ij}(t), \tag{5.13}$$

and the initial distribution vector p has entry 1 for the state of $\mathbf{Z}$ where the individual enters the system (conventionally this corresponds to the state representing the root vertex of the CT-DCEG $\mathcal{D}$) and 0 elsewhere. The renewal kernel $Q(t)$ and initial distribution p completely describe the stochastic process $\mathbf{Z}$. □

Observe that if the CT-DCEG has a sink vertex w_∞, this vertex represents an absorbing state in its associated SMP representation.

Example 5.15. *The state transition diagram of the SMP corresponding to the CT-DCEG for infection process in Figure 5.3, excluding its root vertex and the edges emanating from it, is given in Figure 5.5. The edges* $(w_1, w_3, \textit{strain}\,1)$ *and* $(w_1, w_3, \textit{strain}\,2)$ *of the CT-DCEG are combined into a single edge in the state transition diagram of the SMP. In the state transition diagram, node* w_∞ *is an absorbing state and the remaining states are transient.*

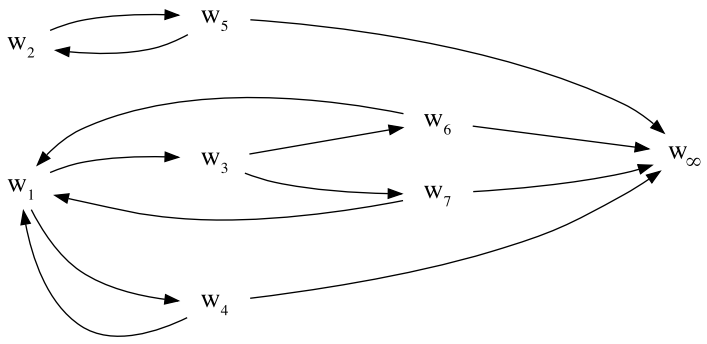


Figure 5.5: State transition diagram of the SMP for the CT-DCEG of the infection process.

Further, if the inference query concerns only a subset of $V(\mathcal{D})$, we can consider a compact version of the SMP containing only vertices/states and edges that are relevant to

the query. This is particularly useful when the CT-DCEG being considered is very large. Such a compact representation is also generally an approximation of the process represented by the CT-DCEG.

Suppose that an inference query concerns only a subset of vertices $V^\dagger \subset V(\mathcal{D})$ of the CT-DCEG $\mathcal{D}$. To use SMP methodologies for this inference query, we construct a compact SMP $\mathbf{Z}^\dagger$ with state space $\mathbf{S}^\dagger = V^\dagger$ as follows. Consider vertices $v_i, v_j \in V^\dagger$. If there exists at least one edge between v_i and v_j in the CT-DCEG $\mathcal{D}$, then the transition probability and holding time distributions are derived as described above. If there is no edge between v_i and v_j in the CT-DCEG $\mathcal{D}$, suppose that there exists exactly one path $\lambda(v_i, v_j)$ from v_i to v_j such that the vertices between v_i and v_j on the path are not in $V^\dagger$. Since $\lambda(v_i, v_j)$ is a path, it visits both v_i and v_j exactly once each. This path can be condensed into an edge e_{ij} from states v_i to v_j in $\mathbf{Z}^\dagger$. The transition probability from state v_i to state v_j and its associated conditional holding time distribution (denoted by variable G) in $\mathbf{Z}^\dagger$ are given by

$$p_{ij} = \prod_{l=1}^{n} \theta_{ijl}, \tag{5.14}$$

$$f_G(t) = (f_{ij1} * \ldots * f_{ijn})(t), \tag{5.15}$$

where n is the number of edges in the path $\lambda(v_i, v_j)$, $*$ represents a convolution and f_{ijl} is the probability density associated with holding time variable H_{ijl}, $1 \leq l \leq n$. Note that in most cases convolutions of probability distributions cannot be solved analytically and will have to be handled numerically.

If there are multiple paths from v_i to v_j in the CT-DCEG $\mathcal{D}$, we first obtain the holding time densities and the transition probabilities for the condensed edges created from each of these paths. We can then combine them into a single edge as described in Equations 5.11 and 5.10.

Figure 5.6: Path from vertex v_i to vertex v_j via vertex v_k.

Finally, we discuss how an SMP representation can be obtained for a CT-DCEG with time-invariant covariates. This process is similar to the one described above for constructing a compact SMP. Suppose that the CT-DCEG $\mathcal{D}$ has vertices representing time-invariant covariates denoted by $V_F(\mathcal{D}) \subset V(\mathcal{D})$. For a valid SMP, we need to define a conditional holding time distribution for each edge in the SMP. The state space $\mathbf{S}$ can be

any proper subset of $V(\mathcal{D})$ so long as a holding time distribution can be defined for each edge. For instance, consider $v_i, v_j \in \mathbf{S}$ such that edge $(v_i, v_j, \cdot) \notin E(\mathcal{D})$ but there exists a directed path $\lambda(v_i, v_j)$ from v_i to v_j in $\mathcal{D}$ through a vertex $v_k \notin \mathbf{S}$ (see Figure 5.6). Further assume that one of v_i and v_k represents a time-invariant covariate. Without loss of generality, assume this is vertex v_i. Thus edge e_{kj} has a conditional holding time variable H_{kj} whereas edge e_{ik} does not have any holding time associated with it. The transition probabilities and conditional holding time distribution along the edge v_i to v_j are then obtained as described in Equations 5.14 and 5.15 respectively.

Note here that a compact representation and incorporation of time-invariant covariates into the SMP representation may only be obtained when there does not exist a directed edge between a vertex v_k and v_l in $\mathcal{D}$ such that $v_l \in \mathbf{S}$, and $v_k \notin \mathbf{S}$ lies on some directed path $\lambda(v_i, v_j)$ in $\mathcal{D}$ for any vertices $v_i, v_j \in \mathbf{S}$ which has been condensed into a single edge in the SMP representation.

Example 5.16 (Infection example continued)**.** *Due to the reason described above, the root vertex w_0 in the CT-DCEG of the infection process in Figure 5.3, representing the time-invariant covariate of an individual's living arrangements, cannot be incorporated into its SMP representation shown in Figure 5.5.*

The difference between SMPs and CT-DCEGs is two-fold. Firstly, much like Markov processes, the states and transitions of an SMP are typically defined at the beginning of the analysis. Of course, if deemed necessary, states and transitions can be added or removed from an SMP through an iterative process. However, this process can be cumbersome. In contrast, a CT-DCEG construction begins by first eliciting – from the domain experts or domain literature – the event tree describing the process. Through the identification of stages and positions, we are then able to identify the vertex and edge sets of the graph of the CT-DCEG. Secondly, by allowing multiple edges (with the same directionality) between vertices, the CT-DCEG is able to express more information about the conditional holding times of the different events associated with the edges that lead to the same outcome.

Nonetheless, an SMP representation allows us to easily answer queries such as first passage times and recurrent visits to states for which well-developed methodologies already exist in the semi-Markov literature; see for example Weiss and Zelen (1965), Janssen and Manca (2006), and Barbu and Limnios (2008) for the more commonly used frequentist SMP methods, and Butler and Huzurbazar (2000), Epifani et al. (2014), and Warr and Woodfield (2019) for a Bayesian treatment of SMPs. The alternative SMP representation, particularly when the representation is approximated by merging edges and vertices, enables us to get a macro-level view of the process modelled by the CT-DCEG. The construction of the CT-DCEG may also be used as the preliminary step in an SMP analysis to elicit the required

states and transitions for the SMP (Barclay, 2014). Further, as we shall demonstrate in Section 5.6, the semi-Markov representation is an integral part of our proposed CT-DCEG probability propagation scheme.

5.5 Unrolling a CT-DCEG

We next discuss the process of unrolling a CT-DCEG. This is analogous to unrolling a discrete time DCEG (Collazo, 2017) or a DBN (Kjærulff, 1992). Being able to unroll a dynamic process is particularly useful for performing inference as we shall see in Section 5.6.

In a CT-DCEG $\mathcal{D}$, say that $w \preceq w'$ if the shortest walk from the root vertex w_0 to w contains fewer or the same number of edges as the shortest walk from w_0 to w', for $w_0, w, w' \in V(\mathcal{D})$.

Definition 5.17 (Cyclic Edge)**.** *An edge $e = (w', w, \cdot)$ from a vertex w' to another vertex w is said to be a cyclic edge if $w \preceq w'$.*

Note here that loops which are edges of the form $e = (w, w, \cdot)$ are also cyclic edges. Next we define *passage-slices* which are analogous to time-slices in discrete time processes. According to Kjærulff (1992), dynamic models for a complex system may be defined as a sequence of submodels such that each submodel represents the state of the entire system at a specific time point or during a specific time interval. Each such time point or time interval is then referred to as a time-slice. Thus for a complex process described by a collection of random variables, the dynamic model represents repeated observations of these random variables and any given time-slice represents only a single observation of each of these random variables.

In the first instance it might appear appealing to describe passage-slices within a CT-DCEG using the concept of a stopped process. For a stochastic process $\mathbf{X}$, its corresponding stopped process $\mathbf{X}^t$ is given by the development of $\mathbf{X}$ up to the stopping time t for some $t > 0$. However, different walks – all starting from the root vertex – may evolve on very different time scales. This is dependent on the holding time distributions along the edges making up the walk. This makes it difficult, if not impossible in most cases, to choose a sequence of times $t_1, t_2, t_3, \ldots$ such that none of the possible walks that can be traversed by any individual in each of the time intervals $[0, t_1)$, $[t_1, t_2)$ and so on contain not more than one occurrence of the same type of event, and such that the set of walks in the time intervals $[t_1, t_2)$, $[t_2, t_3)$ and so on are equivalent.

So we attempt to define passage-slices directly from the topology of the CT-DCEG. This adds to the convenience of being able to draw the unrolled process and do a preliminary visual analysis of the unrolled graph by only using the coloured graph of the CT-DCEG

without having to reference the estimated parameters of the underlying process. We present such a description of a passage-slice below.

Definition 5.18 (Passage-Slice)**.** *The first passage-slice $P(1)$ of a CT-DCEG $\mathcal{D}$ is the subgraph of the graph of the CT-DCEG obtained by deleting its cyclic edges. The subsequent passage-slices $P(k)$ are identical for $k = \{2, 3, \ldots\}$. Each passage-slice $P(k)$ is a collection of connected components, each of which is a subgraph of the graph of the CT-DCEG. Each subgraph is rooted at a vertex into which a cyclic edge from first passage-slice $P(1)$ enters and it contains all vertices and edges on walks starting from this vertex until another cyclic edge or the sink vertex is reached along each of the walks.*

It follows from the definition of passage-slices that consecutive passage-slices are connected by the set of cyclic edges. The first passage-slice of a CT-DCEG is isomorphic to the CEG of the invariant subtree of its underlying event tree with the cyclic edges removed (see Section 5.3.3). Similarly, each connected component of passage-slice $P(k)$, $k = \{2, 3, \ldots\}$ is isomorphic to the CEG of a repeating subtree, again with the cyclic edges removed. Also observe here that events that occur only once (e.g. typically those associated with time-invariant covariates) in the process are only part of the invariant subtree and not the repeating subtrees.

For a CT-DCEG $\mathcal{D}$, denote its set of cyclic edges by $\epsilon \subset E(\mathcal{D})$. Any finite CT-DCEG can be "unrolled" by connecting its passage-slices with the cyclic edges. To begin with, we consider unrolling the graph of the CT-DCEG from the first passage-slice $P(1)$ up to the desired passage-slice $P(k)$, $k > 1$. The process is straightforward. Beginning with the first passage-slice, we connect the consecutive passage-slices with the cyclic edges and collect all the terminating walks into a common sink vertex w_∞. Additionally, all the leaves of the final passage-slice $P(k)$ are also collected into the sink vertex. Unrolling in this way gives us a CEG (evolving in continuous time) of the process from the first to the kth passage-slice. We denote this as the CEG $C_{P(1:k)}$. Figure 5.7 shows a diagrammatic representation of the unrolling of a CT-DCEG from the first to the third passage-slices. Note that such a CEG is generally not square-free (see Section 4.4).

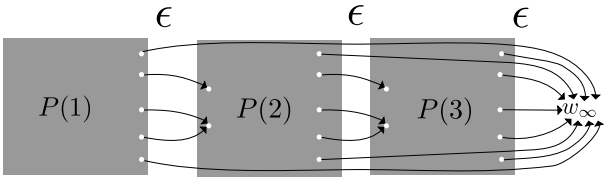


Figure 5.7: Unrolled CT-DCEG from passage-slices 1 to 3, i.e. $C_{P(1:3)}$.

Observe that the unrolled graph of a CT-DCEG generalises the original CT-DCEG graph as it allows for different distributions of its parameters across its passage-slices.

Example 5.19 (Infection example continued)**.** *For the graph of the CT-DCEG in Figure 5.3, we have* $w_1 \prec w_4$, $w_1 \prec w_6$, $w_1 \prec w_7$ *and* $w_2 \prec w_5$. *Thus the edges* $e_{4,1}$, $e_{6,1}$, $e_{7,1}$ *and* $e_{5,2}$ *are cyclic edges. Figure 5.8 shows the graph of* $C_{P(1:2)}$, *i.e. the graph the CT-DCEG unrolled to the first two passage-slices, for the infection process.*

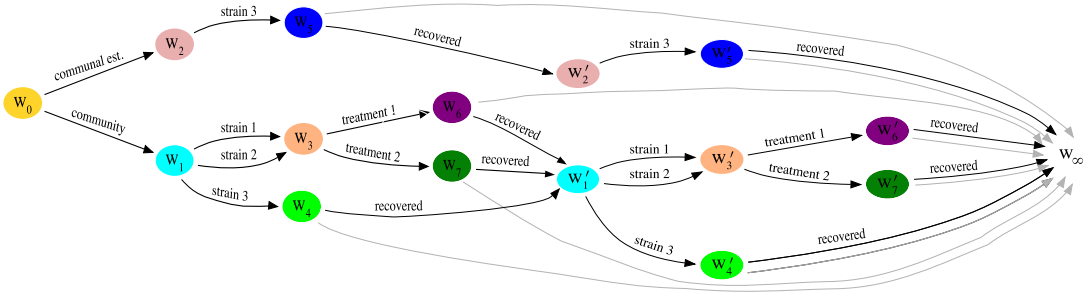
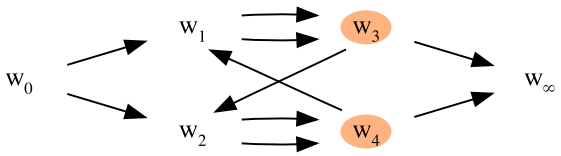


Figure 5.8: $C_{P(1:2)}$ for the infection process. The shaded edges represent hospitalisations.

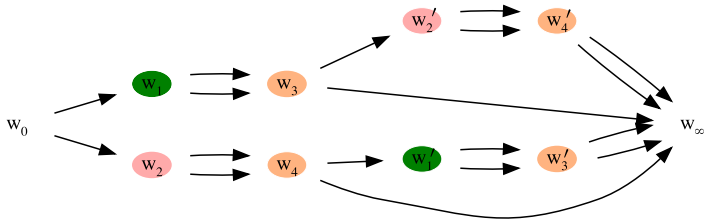
The CEG of an unrolled process need not be a minimal CEG. Recall that the graph of a CEG is said to be minimal when each of its position sets are represented by exactly one vertex. Consider the example below.

Example 5.20. *Consider the CT-DCEG* $\mathcal{D}$ *in Figure 5.9(a). The vertex colouring for the vertices representing singleton stage sets has been suppressed. The vertices* w_3 *and* w_4 *are in the same stage but not the same position as their infinite rooted subtrees are not isomorphic. Figure 5.9(b) shows the CEG* $C_{P(1:2)}$ *of the CT-DCEG unrolled from the first to the second passage-slice. In this CEG, the vertices* w'_2 *and* w'_4 *represent vertices* w_2 *and* w_4 *respectively in the second passage-slice. Here, they are in the same stage and also the same position as they have isomorphic finite rooted subtrees with respect to the CEG. Hence, to obtain the minimal representation of this CEG we would contract vertices* w'_2 *and* w'_4 *into a single vertex.*

A CT-DCEG can also be unrolled from the kth passage-slice $P(k)$ to the $(k + l)$th passage-slice $P(k + l)$ for $k, l \in \mathbb{N}$. Recall that the first passage-slice is the only one guaranteed to contain only one connected component. The remaining passage-slices will have as many connected components as we have repeating subtrees. These passage-slices can



(a)



(b)

Figure 5.9: (a) A CT-DCEG; (b) CEG $\mathcal{C}_{P(1:2)}$ of the unrolled CT-DCEG.

still be unrolled as before but the graph thus obtained does not necessarily correspond to any CEG model. This is because when we have more than one connected component, we will have more than one vertex which has no parents and no incoming edges. The following steps transform this graph denoted below as $G = (V(G), E(G))$ into a single CEG or a collection of CEGs:

- If after removing the edges of the form $(w, w_\infty, \cdot) \in E(G)$ entering the single sink vertex w_∞ the graph G decomposes into two or more connected components, we can then transform each connected component $G_i = (V(G_i), E(G_i))$ into a CEG with graph $G_i^* = (V(G_i^*), E(G_i^*))$ such that

$$V(G_i^*) = V(G_i) \cup \{w_\infty\}$$
$$E(G_i^*) = E(G_i) \cup \{(w, w_\infty, \cdot) \,|\, w \in V(G_i), (w, w_\infty, \cdot) \in E(G)\}.$$

  In other words, each connected component gets its own sink vertex.

- The graph of each connected component is reduced to its minimal representation by merging vertices which are in the same position.

Example 5.21 (Example 5.20 continued)**.** *We consider the graph of the CT-DCEG in Figure 5.9(a) unrolled from passage-slice $P(k)$ to passage-slice $P(k+1)$, for $k \in \mathbb{N}$. By adding another sink vertex w'_∞, the graph can be represented as two distinct CEGs. The graphs of the two CEGs thus obtained are shown in Figure 5.10. The vertex colourings show the non-singleton stages within each CEG.*

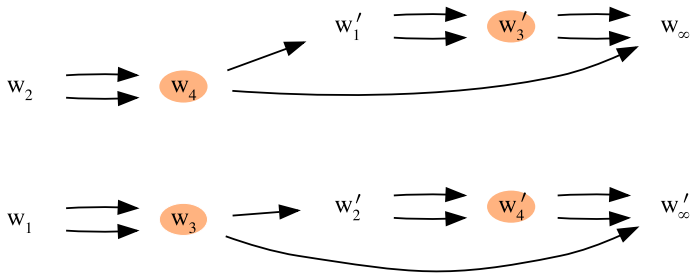
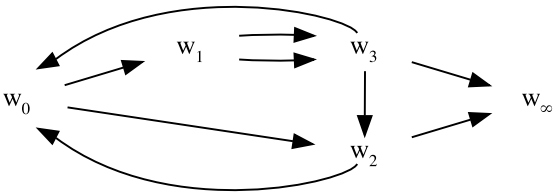


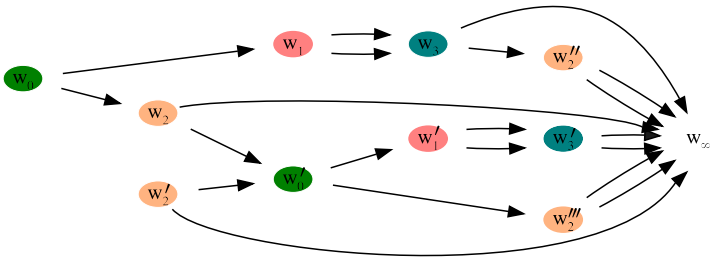
Figure 5.10: Graph of the CT-DCEG in Figure 5.9(a) unrolled from passage-slices k to $k+1$.

Example 5.22. *We consider another example where the graph of the unrolled CT-CEG does not directly decompose into multiple CEGs. Consider the CT-DCEG with the graph shown in Figure 5.11(a). Here again the vertex colouring for the vertices representing singleton stage sets has been suppressed. The graph of the CT-DCEG unrolled from passage-slices k to $k+1$, for $k \in \mathbb{N}$ is shown in Figure 5.11(b). We obtain a minimal representation of this graph by merging w_2 with w'_2, and w''_2 with w'''_2. The graph of the resultant CEG is given in Figure 5.11(c).*

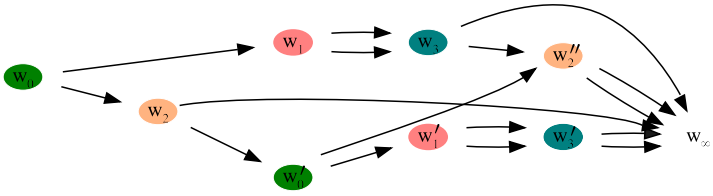
We note that at first glance it might appear inappropriate to apply the concept of a position in a graph that is not already a CEG. However, it might be more convincing to view an unrolled CT-DCEG from passage-slices k to $k+l$ as a subgraph of the minimal CEG obtained by unrolling the same CT-DCEG from passage-slices 1 to $k+l$. A CT-DCEG unrolled from the first passage-slice is necessarily a CEG as described above. We would then expect the subgraph of the graph of a CEG to also represent a CEG or decompose into multiple CEGs.



(a)



(b)



(c)

Figure 5.11: (a) A CT-DCEG; (b) graph of CT-DCEG unrolled from passage-slices k to $k+1$; (c) minimal representation of this graph.

Finally note that we can also unroll just a single passage-slice. The graph of a CT-DCEG unrolled at passage-slice $P(k)$ is not isomorphic to the graph of $P(k)$ itself as the former also contains the cyclic edges emanating from the leaves of $P(k)$.

5.6 Probability Propagation in CT-DCEGs

We now consider probability propagation through a CT-DCEG. Recall that a review of the CEG propagation algorithm which does not consider holding times on the transitions was presented in Section 3.3. This propagation algorithm can be used to propagate intrinsic evidence $\mathscr{E}$ through a CEG whenever the conditional holding times are non-discriminatory (see Section 5.1) or we do not have any temporal evidence. In this section, we explore how observations of the unconditional holding times at the vertices of a CEG may lead to dependence between the conditional holding time distributions and the transition probability distribution at that vertex. This happens precisely when the unconditional holding time distributions on the edges emanating from a given vertex are not equivalent. In other words, if all the edges emanating from a given vertex have the same conditional holding time distribution, any observation relating to the unconditional holding time at that vertex does not offer any discriminatory information about the transition probability distribution at that vertex. In such cases, the vanilla CEG propagation from Section 3.3 may be used. We shall see in this section how we can extend this propagation algorithm to propagate temporal observations of unconditional holding time distributions when they possess such discriminatory power.

Recall that evidence $\mathscr{E}$ for a CT-DCEG refers to the set of observation of edges or vertices traversed or occupied by an individual in the CT-DCEG. We shall consider only evidence that are positive and are point observations. Further, recall from Section 3.3 that we also allow negative evidence which can be framed as uncertain positive evidence, and we assume that the probabilities associated with the elements in a given set of uncertain positive evidence are equal.

Additionally, a CT-DCEG captures the temporal dynamics of the process being modelled and hence we might observe *temporal evidence*. Let elements of the temporal evidence $\mathscr{T}$ be holding times for vertices in the CT-DCEG. As with the evidence, we shall only consider temporal evidence such that its elements are positive (certain or uncertain as described in Section 3.3) and are point observations. We do not consider here temporal observations of the kind where we observe that an individual was at a certain vertex for a given time interval. However, note that such interval temporal observations are of great interest in continuous time models and we discuss this further in Section 5.8.

Example 5.23 (Infection example continued)**.** *Observing that an individual or a group of*

individuals had a holding time of t days in one of w_6 or w_7 is an instance of uncertain positive temporal evidence. Again, we assume here that this evidence could be for w_6 or w_7 with equal probability. That is, we do not have any information to say that the individual was more likely to have received treatment 1 or treatment 2.

Crucially, we require that the temporal evidence $\mathscr{T}$ is compatible with the evidence in $\mathscr{E}$. For instance, the evidence $\mathscr{E}$ and temporal evidence $\mathscr{T}$ would be incompatible if a vertex v in $\mathscr{E}$ makes all paths passing through another vertex v' impossible and $\mathscr{T}$ contains the unconditional holding time at vertex v'. Further, assume that information contained in $\mathscr{T}$ through the transition times about specific edges traversed or vertices visited is also contained in $\mathscr{E}$.

As discussed earlier for CEGs and BNs, not all kinds of evidence can be propagated. The ones which cannot be propagated are precisely those that destroy the conditional independence structures on which such propagation algorithms rely for performing their local message-passing updates (see Section 3.3). Hence, we still require that the evidence $\mathscr{E}$ is intrinsic as defined in Section 3.3. However, no additional constraints are needed for the temporal evidence $\mathscr{T}$.

Before presenting an extension of the vanilla CEG propagation algorithm such that it incorporates temporal evidence, we first present a novel dynamic propagation scheme to simplify the process of propagation through a dynamic CEG variant.

5.6.1 A Dynamic Propagation Scheme

While in theory dynamic variants of CEGs are modelled on an infinite tree and hence can be thought to continue indefinitely, in practice the availability of data and the feasibility of analysis typically restrict us to a fixed number of passage-slices. Hence, although theoretically we can unroll a CT-DCEG into an infinitely large CEG, for most practical purposes it is sufficient to concentrate on a *window* of a finite number of passage-slices, say from k to $k+l$ represented by the unrolled CEG $C_{P(k:k+l)}$, for $k, l \in \mathbb{N}$. In fact, DBNs also rely on such unrolling for inference and the basic idea of our propagation scheme is analogous to the scheme presented in Kjærulff (1992) for DBNs which employs standard junction tree inference by unrolling the DBN and splitting it into past, current and future models. CTBNs, on the other hand, model each variable as a continuous time Markov process. Inference in CTBNs directly exploits these continuous time Markov processes and typically does not involve any unrolling (see e.g. Nodelman et al. (2002) and Saria et al. (2007)).

We now consider how we can simplify our inferential exercise for a CT-DCEG by splitting the problem into *past, current and future* models based on the passage-slices corresponding to the intrinsic evidence $\mathscr{E}$ and temporal evidence $\mathscr{T}$. For a given CT-DCEG

$\mathcal{D}$ with known conditional transition probabilities and holding time distributions, suppose that the intrinsic evidence $\mathscr{E}$ and temporal evidence $\mathscr{T}$ pertain to vertices contained in passage-slices k to $k + l$, for $k, l \in \mathbb{N}$. We can then unroll the CT-DCEG and split it into three models as follows:

- The past model is given by the CEG $C_{P(1:k-1)}$ of the unrolled CT-DCEG.
- The current model is given by the CEG or the collection of CEGs associated with the graph of the CT-DCEG unrolled from passage-slices k to $k + l$.
- The future model is given by an updated CT-DCEG $\mathcal{D}^*$ as described in Section 5.6.4.

The main inferential task lies in propagating the intrinsic evidence $\mathscr{E}$ and temporal evidence $\mathscr{T}$ through the current model. However, new evidence observed for the current model is likely to significantly affect the conditional transition probabilities in the past and future models as well. Propagating the new evidence through vertices that precede the vertices for which evidence has been observed is often referred to as *backward smoothing*. The inference concerning vertices which succeed the vertices for which evidence has been observed is known as *forecasting*. For this reason, the past model may also be referred to as the backward smoothing model and the future model as the forecasting model. However, as stated in Kjærulff (1992), these terms are slightly imprecise as the current model most likely also contains vertices for which we perform backward smoothing and/or forecasting. We first address propagation through the current model before discussing propagation through the past and future models.

5.6.2 Propagation Through the Current Model

Consider a CT-DCEG $\mathcal{D}$ with known conditional transition probabilities and holding time distributions. Let $\mathscr{E}$ denote the intrinsic evidence and $\mathscr{T}$ the temporal evidence which pertain to events contained within the passage-slices from k to $k + l$, for $k, l \in \mathbb{N}$. For simplicity, we shall assume here that the graph of this CT-DCEG unrolled from passage-slices k to $k + l$ represents the graph of a single CEG. Denote this CEG by $C_{P(k:k+l)}$. This unrolled CEG is our current model. If the graph instead decomposes into a collection of CEGs, we would need to propagate $\mathscr{E}$ and $\mathscr{T}$ as described below through each of the CEGs.

Intrinsic evidence $\mathscr{E}$ enables us to reduce the possible root-to-sink walks that one could traverse in $C_{P(k:k+l)}$ conditioned on $\mathscr{E}$ to $\Lambda(\mathscr{E})$. Denote the probability of occupying a vertex $w \in V(C_{P(k:k+l)})$ by $p(w) = p(\Lambda(w))$ and the probability of traversing an edge $(w, w', l) \in E(C_{P(k:k+l)})$ by $p(w, w', l) = p(\Lambda(w, w', l) \,|\, \Lambda(w))$. Further, denote by $p^t(w, w', l) = p(H(w, w', l) = t \,|\, \Lambda(w, w', l), \Lambda(w))$ the probability of staying at position w for time t before transitioning along edge (w, w', l).

If we have no temporal evidence, i.e. $\mathscr{T} = \emptyset$ then propagation through the CEG of the current model proceeds as in the vanilla CEG propagation described in Section 3.3. Observe that the parameters of the conditional holding time distributions do not get revised by any (temporal) evidence. Hence, our focus is on updating the conditional transition probabilities in $C_{P(k:k+l)}$ in light of intrinsic evidence $\mathscr{E}$ and the temporal evidence $\mathscr{T}$. The latter is necessitated for the reasons outlined earlier. We consider two cases here: 1) when we have temporal evidence of holding time for at least one vertex in $C_{P(k:k+l)}$; 2) when we have temporal information of the kind where we know the total holding time from some vertex w to some other vertex w' in $C_{P(k:k+l)}$.

Case 1

Consider the minimal $\mathscr{E}$-reduced graph of $C_{P(k:k+l)}$ (see Section 3.3 for definition of an $\mathscr{E}$-reduced graph). To start with, we suppose that we have holding times for all transitions from the root of $C_{P(k:k+l)}$ up to its sink. Clearly, in this case, if we have n holding times, then any root-to-sink walk in $C_{P(k:k+l)}$ with fewer or more than n edges will have associated probability zero. This is an indirect form of negative evidence implied through the positive temporal evidence in $\mathscr{T}$. In order for $\mathscr{E}$ and $\mathscr{T}$ to be consistent with each other, we require that the intrinsic evidence $\mathscr{E}$ will include this information in the form of certain and uncertain positive evidence such that none of the root-to-sink walks in $C_{P(k:k+l)}$ with fewer or more than n edges are part of its $\mathscr{E}$-reduced graph, $C^{\mathscr{E}}_{P(k:k+l)}$.

Like the vanilla CEG propagation algorithm, the propagation algorithm for the current model has two main steps: a backward step to calculate *potentials* and *emphases* for the conditional transition and conditional holding time distributions, and a forward step which updates the conditional transition probabilities. Note that $p(.)$ refers to probabilities in $C_{P(k:k+l)}$ and $\hat{p}(.)$ to the updated probabilities in $C^{\mathscr{E}}_{P(k:k+l)}$.

Denote by $E(w)$ all the edges emanating from vertex $w \in V(C_{P(k:k+l)})$. Let $V(-1) = \{w \in V(C_{P(k:k+l)}) \,|\, \forall (w, w', \cdot) \in E(w), w' = w_\infty\}$, i.e. $V(-1)$ contains vertices all of whose outgoing edges terminate in w_∞ in $C_{P(k:k+l)}$. The algorithm proceeds as follows.

1. For each edge $e = (w, w_\infty, l) \in E(w)$ for $w \in V(-1)$, if $(w, w_\infty, l) \in E(C^{\mathscr{E}}_{P(k:k+l)})$ set the *t-potential* (conditional transition potential) $\tau_e(w_\infty \,|\, w)$ and *h-potential* (conditional holding time potential) $\tau_e^{t_w}(w_\infty \,|\, w)$ as

$$\tau_e(w_\infty \,|\, w) = p(w, w_\infty, l),$$
$$\tau_e^{t_w}(w_\infty \,|\, w) = p^{t_w}(w, w_\infty, l),$$

where t_w denotes the holding time at w. If $(w, w_\infty, l) \notin E(C^{\mathscr{E}}_{P(k:k+l)})$, set both potentials

as zero. Now set the *t-emphasis* $\Phi(w)$ and *h-emphasis* $\Phi^{t_w}(w)$ as follows

$$\Phi(w) = \sum_{e\in E(w)} \tau_e(w_\infty \mid w),$$
$$\Phi^{t_w}(w) = \sum_{e\in E(w)} \tau_e(w_\infty \mid w)\, \tau_e^{t_w}(w_\infty \mid w).$$

We now say that the sink w_∞ and all the positions in $V(-1)$ are *accommodated*.

2. For an edge $e = (w, w', l) \in E(w)$ for $w \in V(C_{P(k:k+l)})$ such that all of w's children are accommodated, set the t-potential and h-potential as

$$\tau_e(w' \mid w) = p(w, w', l)\, \Phi(w'),$$
$$\tau_e^{t_w}(w' \mid w) = p^{t_w}(w, w', l),$$

if $(w, w', l) \in E(C^{\mathscr{E}}_{P(k:k+l)})$ and zero otherwise. Set the emphases as

$$\Phi(w) = \sum_{e\in E(w)} \tau_e(w' \mid w),$$
$$\Phi^{t_w}(w) = \sum_{e\in E(w)} \tau_e(w' \mid w)\, \tau_e^{t_w}(w' \mid w).$$

Position w is said to be accommodated when the potentials and emphases are calculated for all $e \in E(w)$.

3. For all $w \in V(C_{P(k:k+l)})$ and for edge $e = (w, w', l) \in E(w)$, the updated conditional transition probabilities are given by

$$\hat{p}(e) = \begin{cases} \dfrac{\tau_e(w' \mid w)\, \tau_e^{t_w}(w' \mid w)}{\Phi^{t_w}(w)}, & \text{if } e \in E(C^{\mathscr{E}}_{P(k:k+l)}) \\ 0, & \text{if } e \notin E(C^{\mathscr{E}}_{P(k:k+l)}). \end{cases}$$

The edges of $C^{\mathscr{E}}_{P(k:k+l)}$ are populated with the non-zero conditional transition probabilities $\hat{p}(.)$ and their associated holding time distributions are inherited from $C_{P(k:k+l)}$.

As we worked with the CEG of the unrolled CT-DCEG graph, our algorithm above is essentially a propagation algorithm for CEGs with arbitrary holding time distributions on its edges. Observe further that we do assume in our algorithm that the conditional holding time distributions are continuous. Hence, this algorithm is equally applicable to the case where our process evolves in discrete time and where we have explicit modelling of conditional holding times. We prove the results in our algorithm below.

Theorem 5.24. *For a CT-DCEG $\mathcal{D}$, intrinsic evidence $\mathscr{E}$ and temporal evidence $\mathscr{T}$, suppose that its current model is given by a single CEG $C_{P(k:k+l)}$, for $k, l \in \mathbb{N}$. Suppose that $\mathscr{T}$ which contains holding times of all transitions for the realised, but partially unobserved,*

root-to-sink walk in $C_{P(k:k+l)}$. Let the $\mathscr{E}$-reduced graph of $C_{P(k:k+l)}$ be denoted by $C^{\mathscr{E}}_{P(k:k+l)}$. The conditional transition probabilities in $C^{\mathscr{E}}_{P(k:k+l)}$ are obtained as below:

$$\hat{p}(e) = \begin{cases} \dfrac{\tau_e(w' \mid w)\, \tau_e^{t_w}(w' \mid w)}{\Phi^{t_w}(w)}, & \text{if } e \in E(C^{\mathscr{E}}_{P(k:k+l)}) \\ 0, & \text{if } e \notin E(C^{\mathscr{E}}_{P(k:k+l)}) \end{cases}$$

where $w \in V(C_{P(k:k+l)})$ and edge $e = (w, w', l) \in E(w)$. The potentials and emphases are as defined in the algorithm above.

Proof. The updated transition probability for edge (w, w', l) given the intrinsic evidence $\mathscr{E}$ and the possible unconditional holding time at vertex w given in the temporal evidence $\mathscr{T}$, denoted here by t_w, can be written as

$$\begin{aligned} \hat{p}(w, w', l) &= p(\Lambda(w, w', l) \mid \mathscr{E}, H(w) = t_w, \Lambda(w)) \\ &= \frac{p(\Lambda(w, w', l), \mathscr{E}, H(w) = t_w, \Lambda(w))}{p(\mathscr{E}, H(w) = t_w, \Lambda(w))} \end{aligned} \tag{5.16}$$

Note that t_w is the *possible* unconditional holding time at vertex w as $\mathscr{T}$ gives the holding times at the various transitions. Vertices which are i edges away from the root of $C^{\mathscr{E}}_{P(k:k+l)}$ might all correspond with the holding time associated with ith transition in $\mathscr{T}$.

Let $\mathcal{T}$ denote the event tree underlying the unrolled CEG $C_{P(k:k+l)}$ and $\mathcal{T}^*$ – a subtree of $\mathcal{T}$ – denote the event tree underlying $C^{\mathscr{E}}_{P(k:k+l)}$. Recall that a position set contains a set of vertices in the event tree and that each position set has a single representative vertex in the CEG. So a vertex $w \in V(C_{P(k:k+l)})$ represents a set of vertices denoted by $V(w)$ in the underlying event tree $\mathcal{T}$. Further, we can partition $V(w)$ into two sets $V_I(w)$ and $V_J(w)$ such that vertices in $V_I(w)$ appear in the event tree $\mathcal{T}^*$ while vertices $V_J(w)$ do not. For each edge (w, w', l) in $C_{P(k:k+l)}$, there exists an edge (v, v', l) in $\mathcal{T}$ for every $v \in V(w)$. Hence, we can write $\Lambda(w)$ and $\Lambda(w, w', l)$ in $C_{P(k:k+l)}$ as

$$\begin{aligned} \Lambda(w) &= \cup_{v \in V(w)} \Lambda(v), \\ \Lambda(w, w', l) &= \cup_{v \in V(w)} \Lambda(v, v', l). \end{aligned}$$

Further, we have that for edge $(v, v', l) \in E(\mathcal{T})$ where $v \in V(w)$ and $v' \in V(w')$,

$$\begin{aligned} p(v, v', l) &= p(w, w', l), \\ p^{t_v}(v, v', l) &= p^{t_w}(w, w', l), \end{aligned}$$

where $t_v = t_w$. We can rewrite Equation 5.16 as

$$\hat{p}(w, w', l) = \frac{p(\mathscr{E}, \cup_{v \in V(w)} \{\Lambda(v, v', l), H(v) = t_v, \Lambda(v)\})}{p(\mathscr{E}, \cup_{v \in V(w)} \{H(v) = t_v, \Lambda(v)\})}, \tag{5.17}$$

to be evaluated on the tree $\mathcal{T}$. There is no directed path from v_i and v_j for $v_i, v_j \in V(w), i \neq j$ as their subtrees are isomorphic in $\mathcal{T}$. Hence we have that $\Lambda(v_i) \cap \Lambda(v_j) = \emptyset$ and $\Lambda(v_i, v_i', l) \cap \Lambda(v_j, v_j', l) = \emptyset$. So we can write Equation 5.17 as

$$\hat{p}(w, w', l) = \frac{\sum_{v \in V(w)} p(\mathscr{E}, \Lambda(v, v', l), H(v) = t_v, \Lambda(v))}{\sum_{v \in V(w)} p(\mathscr{E}, H(v) = t_v, \Lambda(v))}. \tag{5.18}$$

For $v \in V_J(w)$ we have that ${\mathcal{T}^*}_\Lambda \cap \Lambda(v) = \emptyset$. Also, since the vertices in the same position are exchangeable, we can write Equation 5.18 as

$$\begin{aligned}
\hat{p}(w, w', l) &= \frac{\sum_{v \in V_I(w)} p(\mathscr{E}, \Lambda(v, v', l), H(v) = t_v, \Lambda(v))}{\sum_{v \in V_I(w)} p(\mathscr{E}, H(v) = t_v, \Lambda(v))} \\
&= \frac{p(H(v_i) = t_{v_i} \,|\, \mathscr{E}, \Lambda(v_i, v_i', l), \Lambda(v_i))}{p(\mathscr{E}, H(v_i) = t_{v_i} \,|\, \Lambda(v_i))} \times \frac{p(\mathscr{E}, \Lambda(v_i, v_i', l) \,|\, \Lambda(v_i)) \sum_{v \in V_I(w)} p(\Lambda(v))}{\sum_{v \in V_I(w)} p(\Lambda(v))} \\
&= \frac{p(H(v_i) = t_{v_i} \,|\, \mathscr{E}, \Lambda(v_i, v_i', l), \Lambda(v_i))}{p(\mathscr{E}, H(v_i) = t_{v_i} \,|\, \Lambda(v_i))} \times p(\mathscr{E}, \Lambda(v_i, v_i', l) \,|\, \Lambda(v_i))
\end{aligned} \tag{5.19}$$

for any $v_i \in V_I(w)$.

The proofs for $\Phi(w) = p(\mathscr{E} \,|\, \Lambda(v))$ and $\tau_e(w' \,|\, w) = p(\mathscr{E}, \Lambda(v, v', l)) \,|\, \Lambda(v))$ where $e = (w, w', l) \in C^{\mathscr{E}}_{P(k:k+l)}$ and $v \in V_I(w)$, follow exactly as given in Thwaites et al. (2008). We present these below for completeness.

Proof for $\Phi(w) = p(\mathscr{E} \,|\, \Lambda(v))$ by induction

Base case: Consider the vertices $w \in W(-1)$. We have for any $v \in V_I(w)$

$$\begin{aligned}
\Phi(w) &= \sum_{e \in E(w)} \tau_e(w_\infty | w) \\
&= \sum_{e \in E(w)} p(w, w_\infty, l) \\
&= \sum_{e \in E(v)} p(v, v_{\text{leaf}}, l) \\
&= p(\mathscr{E} \,|\, \Lambda(v)).
\end{aligned}$$

Generalisation: Next, we consider any vertex $w \in V(C_{P(k:k+l)})$ such that all the vertices $\{w'\}$ into which edges from w (i.e. edges in $E(w)$) terminate have $\Phi(w') = p(\mathscr{E} \,|\, \Lambda(v'))$ for any

$v' \in V_I(w')$. Then we have for any $v \in V_I(w)$

$$\begin{aligned}\Phi(w) &= \sum_{e \in E(w)} \tau_e(w'|w) \\ &= \sum_{e \in E(w)} p(w, w', l)\, \Phi(w') \\ &= \sum_{e \in E(v)} p(v, v', l)\, p(\mathscr{E} \,|\, \Lambda(v')) \\ &= \sum_{e \in E(v)} p(\Lambda(v, v', l) \,|\, \Lambda(v))\, p(\mathscr{E} \,|\, \Lambda(v')).\end{aligned}$$

However, in an event tree, we have that $\Lambda(v') = \Lambda(v, v', l) \subset \Lambda(v)$. So we can write $\Phi(w)$ as

$$\begin{aligned}\Phi(w) &= \sum_{e \in E(v)} p(\Lambda(v, v', l), \Lambda(v') \,|\, \Lambda(v))\ p(\mathscr{E} \,|\, \Lambda(v'), \Lambda(v, v', l), \Lambda(v)) \\ &= \sum_{e \in E(v)} p(\mathscr{E}, \Lambda(v'), \Lambda(v, v', l) \,|\, \Lambda(v)) \\ &= \sum_{e \in E(v)} p(\mathscr{E}, \Lambda(v, v', l) \,|\, \Lambda(v)) \\ &= p(\mathscr{E}, \Lambda(v) \,|\, \Lambda(v)) \\ &= p(\mathscr{E} \,|\, \Lambda(v)).\end{aligned}$$

Proof for $\tau_e(w' \,|\, w) = p(\mathscr{E}, \Lambda(v, v', l) \,|\, \Lambda(v))$

For any $v \in V_I(w)$ and edge $e = (w, w', l) \in E(w)$, we have that

$$\begin{aligned}\tau_e(w' \,|\, w) &= p(w, w', l)\, \Phi(w') \\ &= p(v, v', l)\, p(\mathscr{E} \,|\, \Lambda(v')) \\ &= p(\Lambda(v, v', l) \,|\, \Lambda(v))\, p(\mathscr{E} \,|\, \Lambda(v')).\end{aligned}$$

Now using the same reasoning as for the generalisation step in the previous proof,

$$\tau_e(w' \,|\, w) = p(\mathscr{E}, \Lambda(v, v', l) \,|\, \Lambda(v)).$$

We now present the proofs for $\Phi^{t_w}(w) = p(\mathscr{E}, H(v) = t_v \,|\, \Lambda(v))$ and $\tau_e^{t_w}(w'|w) = p(H(v) = t_v|\mathscr{E}, \Lambda(v, v', l), \Lambda(v))$ where $e = (w, w', l) \in C^{\mathscr{E}}_{P(k:k+l)}$ and any $v \in V_I(w)$.

Proof for $\Phi^{t_w}(w) = p(\mathscr{E}, H(v) = t_v \,|\, \Lambda(v))$ by induction

Base case: Consider the vertices $w \in W(-1)$. We have for any $v \in V_I(w)$ and $t_v = t_w$,

$$\begin{aligned}
\Phi^{t_w}(w) &= \sum_{e \in E(w)} \tau_e(w_\infty|w)\, \tau_e^{t_w}(w_\infty|w) \\
&= \sum_{e \in E(w)} p(w, w_\infty, l)\, p^{t_w}(w, w_\infty, l) \\
&= \sum_{e \in E(v)} p(v, v_{\text{leaf}}, l)\, p^{t_v}(v, v_{\text{leaf}}, l) \\
&= \sum_{e \in E(v)} p(\Lambda(v, v_{\text{leaf}}, l) \,|\, \Lambda(v))\, p(H(v, v_{\text{leaf}}, l) = t_v \,|\, \Lambda(v, v_{\text{leaf}}, l), \Lambda(v)) \\
&= \sum_{e \in E(v)} p(H(v, v_{\text{leaf}}, l) = t_v, \Lambda(v, v_{\text{leaf}}, l) \,|\, \Lambda(v)).
\end{aligned}$$

However, we have that $p(H(v, v_{\text{leaf}}, l) = t_v, \Lambda(v, v_{\text{leaf}}, l)|\Lambda(v)) = p(H(v) = t_v, \Lambda(v, v_{\text{leaf}}, l)|\Lambda(v))$ as the holding time distributions depend on the edge traversed. So we can write $\Phi^{t_w}(w)$ as

$$\begin{aligned}
\Phi^{t_w}(w) &= \sum_{e \in E(v)} p(H(v) = t_v, \Lambda(v, v_{\text{leaf}}, l) \,|\, \Lambda(v)) \\
&= p(H(v) = t_v \,|\, \Lambda(v)) \sum_{e \in E(v)} p(\Lambda(v, v_{\text{leaf}}, l) \,|\, \Lambda(v), H(v) = t_v) \\
&= p(H(v) = t_v \,|\, \Lambda(v))\, p(\mathscr{E}, \Lambda(v) \,|\, \Lambda(v), H(v) = t_v) \\
&= p(\mathscr{E}, H(v) = t_v \,|\, \Lambda(v)).
\end{aligned}$$

Generalisation: Now consider any vertex $w \in V(C_{P(k:k+l)})$ such that all the vertices $\{w'\}$ into which edges from w (i.e. edges in $E(w)$) terminate have $\Phi^{t_{w'}}(w') = p(\mathscr{E}, H(v') = t_{v'} \,|\, \Lambda(v'))$ for some $v' \in V_I(w')$. Then we have for any $v \in V_I(w)$ and $t_v = t_w$,

$$\begin{aligned}
\Phi^{t_w}(w) &= \sum_{e \in E(w)} \tau_e(w'|w)\, \tau_e^{t_w}(w'|w) \\
&= \sum_{e \in E(w)} p(w, w', l)\, \Phi(w')\, p^{t_w}(w, w', l) \\
&= \sum_{e \in E(v)} p(v, v', l)\, p(\mathscr{E} \,|\, \Lambda(v'))\, p^{t_v}(v, v', l).
\end{aligned}$$

Again, recall that $\Lambda(v') = \Lambda(v, v', l) \subset \Lambda(v)$. So we can write $\Phi^{t_w}(w)$ as

$$\begin{aligned}
\Phi^{t_w}(w) &= \sum_{e \in E(v)} p(\Lambda(v, v', l), \Lambda(v') \,|\, \Lambda(v))\, p(\mathscr{E} \,|\, \Lambda(v'), \Lambda(v, v', l), \Lambda(v))\, p^{t_v}(v, v', l) \\
&= \sum_{e \in E(v)} p(\mathscr{E}, \Lambda(v'), \Lambda(v, v', l) \,|\, \Lambda(v))\, p^{t_v}(v, v', l) \\
&= \sum_{e \in E(v)} p(\mathscr{E}, \Lambda(v, v', l) \,|\, \Lambda(v))\, p^{t_v}(v, v', l) \\
&= \sum_{e \in E(v)} p(\mathscr{E}, \Lambda(v, v', l) \,|\, \Lambda(v))\, p(H(v, v', l) = t_v \,|\, \Lambda(v, v', l), \Lambda(v)).
\end{aligned}$$

Recall that the holding time density is invariant given intrinsic evidence $\mathscr{E}$. So we can write $\Phi^{t_w}(w)$ as

$$\begin{aligned}
\Phi^{t_w}(w) &= \sum_{e \in E(v)} p(\mathscr{E}, \Lambda(v, v', l) \,|\, \Lambda(v))\, p(H(v, v', l) = t_v \,|\, \mathscr{E}, \Lambda(v, v', l), \Lambda(v)) \\
&= \sum_{e \in E(v)} p(\mathscr{E}, H(v, v', l) = t_v, \Lambda(v, v', l) \,|\, \Lambda(v)) \\
&= \sum_{e \in E(v)} p(\mathscr{E}, H(v) = t_v, \Lambda(v, v', l) \,|\, \Lambda(v)) \\
&= p(H(v) = t_v \,|\, \Lambda(v)) \sum_{e \in E(v)} p(\mathscr{E}, \Lambda(v, v', l) \,|\, \Lambda(v), H(v) = t_v) \\
&= p(H(v) = t_v \,|\, \Lambda(v))\, p(\mathscr{E}, \Lambda(v) \,|\, \Lambda(v), H(v) = t_v) \\
&= p(\mathscr{E}, H(v) = t_v \,|\, \Lambda(v)).
\end{aligned}$$

Proof for $\tau_e^{t_w}(w'|w) = p(H(v) = t_v|\mathscr{E}, \Lambda(v, v', l), \Lambda(v))$

For edge $e = (w, w', l) \in E(w)$, any $v \in V_I(w)$ and $t_v = t_w$ we have by definition of $\tau_e^{t_w}(w' \,|\, w)$ and by the invariance of the holding time density given intrinsic evidence $\mathscr{E}$ that

$$\begin{aligned}
\tau_e^{t_w}(w' \,|\, w) &= p^{t_w}(w' \,|\, w) \\
&= p(H(v) = t_v \,|\, \Lambda(v, v', l), \Lambda(v)) \\
&= p(H(v) = t_v \,|\, \mathscr{E}, \Lambda(v, v', l), \Lambda(v)),
\end{aligned}$$

Finally, combining these results for edge $e = (w, w', l) \in E(w)$, any $v \in V_I(w)$ and $t_v = t_w$ enables us to express Equation 5.19 as

$$\begin{aligned}
\hat{p}(w, w', l) &= \frac{p(H(v) = t_v \,|\, \mathscr{E}, \Lambda(v, v', l), \Lambda(v))\, p(\mathscr{E}, \Lambda(v, v', l) \,|\, \Lambda(v))}{p(\mathscr{E}, H(v) = t_v \,|\, \Lambda(v))} \\
&= \frac{\tau_e^{t_w}(w' \,|\, w)\, \tau_e(w' \,|\, w)}{\Phi^{t_w}(w)}.
\end{aligned}$$

This completes the proof. □

The pseudo-code for the above algorithm is given in Algorithm 4. In the algorithm, $V^{-1}(w_i)$ denotes the vertices that have edges terminating in vertex w_i and $E^{-1}(w_i)$ denotes the edges terminating in w_i in the CEG $C_{P(k:k+l)}$. The possible holding time at vertex w_i is denoted by t_i.

Algorithm 4: Current model propagation algorithm

Input : Conditional transition probabilities, holding time distributions and the $\mathscr{E}$-reduced graph for the current model $C_{P(k:k+l)}$, intrinsic evidence $\mathscr{E}$ and temporal evidence $\mathscr{T}$ with holding times for each transition in the realised root-to-sink walk.

Output: Updated conditional transition probabilities.

Denote by $C^{\mathscr{E}}_{P(k:k+l)}$ the $\mathscr{E}$-reduced graph of $C_{P(k:k+l)}$.
Set $A \leftarrow \emptyset$, $B \leftarrow \{w_\infty\}$, $\Phi(w_\infty) \leftarrow 1$.
while $B \neq \{w_0\}$ *(the root vertex)* **do**
  for $w_j \in B$ **do**
    for $w_i \in V^{-1}(w_j)$ **do**
      for $e \in E(w_i) \cap E^{-1}(w_j)$ **do**
        if $e \in \Lambda(\mathscr{E})$ **then**
          $\tau_e \leftarrow p(e) \cdot \Phi(w_j)$, $\tau_e^{t_i} \leftarrow p^{t_i}(e)$
        else
          $\tau_e \leftarrow 0$, $\tau_e^{t_i} \leftarrow 0$
        $A \leftarrow A \cup \{e\}$
      if $E(w_i) \subseteq A$ **then**
        $\Phi(w_i) = \sum_{e \in E(w_i)} \tau_e$
        $\Phi^{t_i}(w_i) = \sum_{e \in E(w_i)} \tau_e \cdot \tau_e^{t_i}$
        $B \leftarrow B \cup \{w_i\}$
    $B \leftarrow B \backslash \{w_j\}$
for $w_i \in V(C_{P(k:k+l)})$ **do**
  for $e \in E(w_i) \cap \Lambda(\mathscr{E})$ **do**
    $\hat{p}(e) = \dfrac{\tau_e \cdot \tau_e^{t_i}}{\Phi^{t_i}(w_i)}$
  for $e \in E(w_i) \backslash \Lambda(\mathscr{E})$ **do**
    $\hat{p}(e) = 0$
return *Updated conditional transition probabilities* $\hat{p}(.)$

If we observe the holding times only up to some vertex $w \neq w_\infty$ in $C_{P(k:k+l)}$, we can still use the above algorithm by setting the h-potential for edges $e = (w', w'', l) \in E(C^{\mathscr{E}}_{P(k:k+l)})$ such that $w \prec w'$ as one. This is equivalent to integrating over the unknown holding time t_w

whenever the holding time at some vertex w is not observed. That is, for any $v \in V_I(w)$,

$$\begin{aligned}\tau_e^{t_w}(w'|w) &= \int_t \tau_e^t(w'|w)\, dt \\ &= \int_t p(H(v) = t \,|\, \mathscr{E}, \Lambda(v, v', l), \Lambda(v))\, dt \\ &= 1 \end{aligned} \tag{5.20}$$

and also, for the h-emphasis, we have

$$\begin{aligned}\Phi^{t_w}(w) &= \int_t \Phi^t(w)\, dt \\ &= \int_t p(\mathscr{E}, H(v) = t \,|\, \Lambda(v))\, dt \\ &= p(\mathscr{E} \,|\, \Lambda(v)) \int_t p(H(v) = t \,|\, \mathscr{E}, \Lambda(v))\, dt \\ &= p(\mathscr{E} \,|\, \Lambda(v)) \\ &= \Phi(w). \end{aligned} \tag{5.21}$$

Thus in this case, the backward propagation exercise until the edges emanating from vertex w are reached, is identical to that in a vanilla CEG. We can further generalise this to set the h-potential to one as above for all vertices where we do not observe a holding time even when they precede vertices for which we observe holding times. In this way, we can also propagate evidence through the vertices representing time-invariant covariates in a CT-DCEG.

Case 2

We now consider the second case where we know the total holding time $t_{w,w'}$ starting from some vertex w until another vertex w' is reached where $w \prec w'$ and $w, w' \in C_{P(k:k+l)}$. Note that all w-to-w' walks in $C^{\mathscr{E}}_{P(k:k+l)}$ are not necessarily of the same length. Denote the set of these walks by $\Lambda(w, w')$. We now estimate the probability of each walk in $\Lambda(w, w')$ given the total time taken for the transition between the two.

We first construct the $\mathscr{E}$-reduced graph $C^{\mathscr{E}}_{P(k:k+l)}$ and perform the vanilla CEG propagation with only the intrinsic evidence $\mathscr{E}$ as described in Section 3.3. Denote the updated conditional transition probabilities by $\hat{p}(.)$. For each walk in $\lambda_i \in \Lambda(w, w')$, let the random variable $H(\lambda_i(w, w'))$ indicate the time it takes to get from vertex w to w' when the individual traverses the edges given in the walk λ_i. Then $H(\lambda_i(w, w'))$ is a convolution of the holding time densities on the edges in the walk λ_i. Since we observe a total holding time for the transition from vertex w to w', at least one of the edges along each λ_i must have a hold-

ing time distribution. In other words, they cannot all emanate from vertices representing time-invariant covariates. Edges along λ_i that do emanate from vertices representing time-invariant covariates do not contribute to $H(\lambda_i(w, w'))$. The probability that the individual travelled to vertex w' from vertex w along the edges in the walk λ_i given the total transition time of $t_{w,w'}$ can be calculated as

$$\hat{p}(\lambda_i \,|\, H(\lambda_i(w, w')) = t_{w,w'}, \mathscr{E}) = \frac{p(H(\lambda_i(w, w')) = t_{w,w'} \,|\, \lambda_i, \mathscr{E})\hat{p}(\lambda_i, \mathscr{E})}{\sum_{\lambda_k \in \Lambda(w,w')} p(H(\lambda_k(w, w')) = t_{w,w'} \,|\, \lambda_k, \mathscr{E})\hat{p}(\lambda_k, \mathscr{E})}$$

where $\hat{p}(\lambda_i, \mathscr{E})$ is given as

$$\hat{p}(\lambda_i, \mathscr{E}) = \prod_{e \in \lambda_i} \hat{p}(e).$$

This type of inference might be suitable for several domains – most notably medicine (e.g. time between two related sets of symptoms to identify what might be the underlying process causing the symptoms) and law/forensic science (e.g. time between two CCTV footages capturing the suspect to infer what the suspect might have done in the time between the two recordings).

Note here that if the convolution of conditional holding time distributions along each walk $\lambda_i \in \Lambda(w, w')$ are not equivalent, then a dependence between the convolutions of conditional holding time distributions and the distribution of transition probabilities along each λ_i will be induced on observing the holding time from w to w'. Within a non-stratified setting, this may occur even if the conditional holding time distributions for the edges emanating from any given vertex are always equivalent.

Finally, following on the discussion in Section 3.3 we note here that when the conditional transition and holding time distributions are estimated, we can substitute the known conditional transition probabilities with the posterior means of the estimated conditional transition distributions, and the known conditional holding time distributions with the posterior conditional holding time distributions. For instance, consider a conditional holding time distribution $H(e)$ that follows a Weibull distribution with known shape parameter κ and unknown scale parameter π. Suppose that the scale parameter π has a posterior Inverse-Gamma distribution with shape hyperparameter β and scale hyperparameter γ. Then the probability that the observed holding time of variable $H(e)$ is t can be obtained as follows

$$\begin{aligned} p(H(e) = t) &= \int_\pi p(t \,|\, \pi, \kappa)\; p(\pi \,|\, \beta, \gamma)\; d\pi \\ &= \int_\pi \left\{ \frac{\kappa}{\pi}(t)^{\kappa-1} \exp\left(\frac{-t^\kappa}{\pi} \right) \right\} \left\{ \frac{\gamma^\beta}{\Gamma(\beta)}(\pi)^{-\beta-1} \exp\left(\frac{-\gamma}{\pi} \right) \right\} d\pi \\ &= \frac{\kappa(\gamma)^\beta (t)^{\kappa-1}\beta}{(\gamma + t^\kappa)^{\beta+1}}. \end{aligned} \tag{5.22}$$

5.6.3 Backward Smoothing

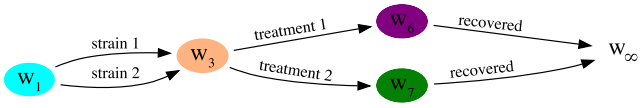
In this section, we discuss backward smoothing. Recall that backward smoothing refers to propagation of information through vertices that precede vertices for which evidence is observed. The current model $C_{P(k:k+l)}$ might contain vertices for which we need to perform backward smoothing and thus, backward smoothing is not restricted to the past model $C_{P(1:k-1)}$. In fact, when the current model contains such vertices, backward smoothing is an implicit part of the propagation algorithm.

Observe that the temporal evidence does not affect the past passage-slices as we do not have any information about the holding times at any vertices in the past passage-slices. While intrinsic evidence might affect the vertices and edges in the past model, this effect need not be propagated through the past model unless we need to update the conditional transition probability distributions or make inferences about the vertices within the past passage-slices. Below we describe how this can be done.

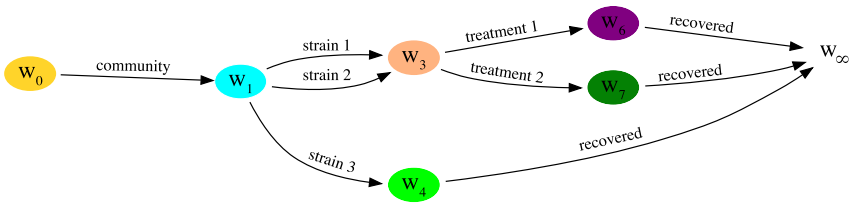
Suppose that our inference query concerns vertices from passage-slices $P(k-j)$ to $P(k-1)$ (for $j \in \mathbb{N}, j < k$) conditional on intrinsic evidence $\mathscr{E}$ and temporal evidence $\mathscr{T}$ which concern vertices from passage-slices $P(k)$ to $P(k+l)$. Denote the CEG of the CT-DCEG unrolled from passage-slices $P(k-j)$ to $P(k-1)$ by $C_{P(k-j:k-1)}$. Since $\mathscr{T}$ does not affect vertices in $C_{P(k-j:k-1)}$, we only need to propagate the intrinsic evidence $\mathscr{E}$ through $C_{P(k-j:k-1)}$. For this we need the $\mathscr{E}$-reduced graph of $C_{P(k-j:k-1)}$. However, vertices and edges in $\mathscr{E}$ do not appear in $C_{P(k-j:k-1)}$. Instead, we obtain the $\mathscr{E}$-reduced graph of $C_{P(k-j:k-1)}$ by deleting the vertices and edges in $C_{P(k-j:k-1)}$ that do no appear on any root-to-sink paths which would connect to $C_{P(k:k+l)}$ if the CT-DCEG were to be unrolled from passage-slices $P(k-j)$ to $P(k+l)$. In simpler words, we delete any vertices and edges in the graph of $C_{P(k-j:k-1)}$ which have zero probability of being visited and traversed when we condition on the intrinsic evidence $\mathscr{E}$. Denote this graph by $C^{\mathscr{E}}_{P(k-j:k-1)}$. The propagation exercise is now reduced to the vanilla CEG propagation algorithm given in Section 3.3.

5.6.4 Forecasting

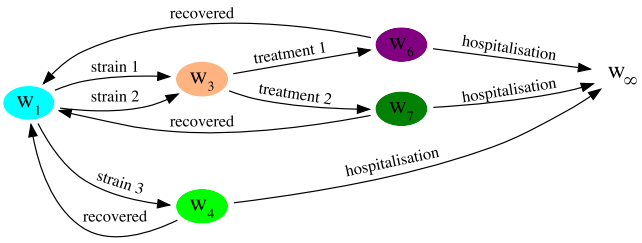
Similar to backward smoothing, the temporal evidence does not affect the future passage-slices for the reasons presented in Section 5.6.3. To obtain the future model, we update the graph of the CT-DCEG $\mathcal{D}$ such that we remove any vertices and edges which have zero probability of being visited and traversed in all future passage-slices $P(i)$ for $i > k+l$ conditioned on intrinsic evidence $\mathscr{E}$. The transition probabilities and holding time distributions on the remaining vertices and edges remain unchanged. Denote this updated CT-DCEG model by $\mathcal{D}^*$. We can now use the CT-DCEG $\mathcal{D}^*$ or its (possibly approximate) SMP representation as described in Section 5.4 for inference queries relating to future passage-slices.



(a)



(b)



(c)

Figure 5.12: $\mathscr{E}$-reduced graphs (a) of current model, (b) past model, and (c) future model for propagating $\mathscr{E}$ and $\mathscr{T}$ as described in Example 5.25.

Example 5.25 (Infection example continued)**.** *Suppose we observe that an individual has been infected by the virus for a second time. Further we observe that they received treatment and recovered. Since the evidence relates to the second passage-slice only, our current model is the CT-DCEG of the infection process unrolled at the second passage-slice. This evidence can be written as* $\mathscr{E} = (\{w_6, w_7\}, \{(w_6, w_\infty, \textit{recovered}), (w_7, w_\infty, \textit{recovered})\})$ *where elements within brackets* $\{\cdot\}$ *represent a set of uncertain evidence. The* $\mathscr{E}$*-reduced graph of the current model, denoted here by* $C^{\mathscr{E}}_{P(2)}$ *is given in Figure 5.12(a). Since the set of root-to-sink paths implied by* $\mathscr{E}$ *in the current model is equivalent to the root-to-sink paths in its* $\mathscr{E}$*-reduced graph, the evidence* $\mathscr{E}$ *is intrinsic. These root-to-sink paths are given below:*

$$\lambda_1 = ((w_1, w_3, \textit{strain 1}), (w_3, w_6, \textit{treatment 1}), (w_6, w_\infty, \textit{recovered}));$$
$$\lambda_2 = ((w_1, w_3, \textit{strain 1}), (w_3, w_7, \textit{treatment 2}), (w_7, w_\infty, \textit{recovered}));$$
$$\lambda_3 = ((w_1, w_3, \textit{strain 2}), (w_3, w_6, \textit{treatment 1}), (w_6, w_\infty, \textit{recovered}));$$
$$\lambda_4 = ((w_1, w_3, \textit{strain 2}), (w_3, w_7, \textit{treatment 2}), (w_7, w_\infty, \textit{recovered})).$$

Assume that the means of the posterior Dirichlet distributions on the conditional transition parameters in the CT-DCEG for the vertices in $C^{\mathscr{E}}_{P(2)}$ *are as described in Table 5.1, and that the Weibull shape parameter and the hyperparameters of the Inverse-Gamma distribution on the Weibull scale parameter for the conditional holding times in the CT-DCEG for the edges in* $C^{\mathscr{E}}_{P(2)}$ *are as given in Table 5.2.*

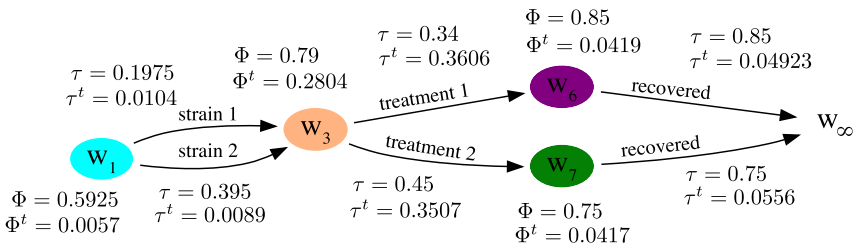
Vertex	Mean posterior transition probabilities	Corresponding edge labels
w_1	(0.25, 0.5, 0.25)	(strain 1, strain 2, strain 3)
w_3	(0.4, 0.6)	(treatment 1, treatment 2)
w_6	(0.85, 0.15)	(recovered, hospitalisation)
w_7	(0.75, 0.15)	(recovered, hospitalisation)

Table 5.1: The mean posterior transition probabilities and the corresponding edge labels in the CT-DCEG for the vertices in $C^{\mathscr{E}}_{P(2)}$.

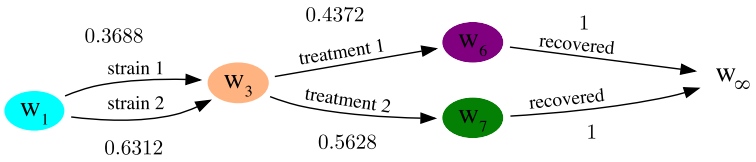
The associated temporal evidence $\mathscr{T}$ *gives us the holding times for the three transitions in the realised but partially unobserved root-to-sink path of* $C^{\mathscr{E}}_{P(2)}$ *as* $t_1 = 35, t_2 = 1$ *and* $t_3 = 13$*, all in days. By propagating* $\mathscr{E}$ *and* $\mathscr{T}$ *through* $C^{\mathscr{E}}_{P(2)}$ *enables us to revise our beliefs of which path the individual might have traversed during their second bout of the infection. Figure 5.13(a) shows the potentials and emphases for the backward step of our propagation algorithm, and Figure 5.13(b) gives the updated transition probabilities for* $C^{\mathscr{E}}_{P(2)}$ *conditioned on* $\mathscr{E}$ *and* $\mathscr{T}$*.*

Edge	Shape parameter	Posterior distribution for scale
$(w_1, w_3, \text{strain 1})$	1	IG(350, 14500)
$(w_1, w_3, \text{strain 2})$	1	IG(720, 42500)
$(w_3, w_6, \text{treatment 1})$	1	IG(425, 520)
$(w_3, w_7, \text{treatment 2})$	1	IG(645, 892)
$(w_6, w_\infty, \text{recovered})$	2	IG(362, 108865)
$(w_7, w_\infty, \text{recovered})$	2	IG(483, 68650)

Table 5.2: The Weibull shape parameter and the posterior Inverse-Gamma distribution for the Weibull scale parameter in the CT-DCEG corresponding to the edges in $\mathcal{C}^{\mathscr{E}}_{P(2)}$.



(a)



(b)

Figure 5.13: (a) Calculation of the potentials and emphases in the backward step of our propagation algorithm; (b) The updated current model with the revised transition probabilities obtained through the forward step of our propagation algorithm.

The probability $\hat{p}(\lambda_i)$ that the individual went along the root-to-sink path λ_i, for $i = 1, 2, 3, 4$ is given in Table 5.3. For comparison, we show the path probabilities in the unrolled CEG $C_{P(2)}$ of the second passage-slice before propagating the intrinsic and temporal evidence ($p(\lambda_i)$) as well as what the path probabilities would have been if we had not corrected them for the temporal evidence and had only used the intrinsic evidence ($p^*(\lambda_i)$). Note that the probabilities $p(\lambda_i)$ for the paths before any intrinsic or temporal evidence has been propagated do not add up to 1.

Path λ_i	$p(\lambda_i)$	$p^*(\lambda_i)$	$\hat{p}(\lambda_i)$
λ_1	0.085	0.142	0.1612
λ_2	0.1125	0.188	0.2076
λ_3	0.17	0.2884	0.2760
λ_4	0.225	0.3816	0.3552

Table 5.3: Path probabilities for path λ_i ($i = 1, 2, 3, 4$) in the current model: $p(\lambda_i)$ before propagating the intrinsic and temporal evidence; $p^*(\lambda_i)$ after propagating intrinsic evidence; $\hat{p}(\lambda_i)$ after propagating both the intrinsic and temporal evidence.

One of the most important features of propagation in a CT-DCEG (like in a CEG) is that evidence typically leads to simplification of the graph through which we need to propagate the evidence. Further, the above example clearly shows how knowing the holding times conveys essential information about the evolution of the process. Our propagation algorithm conveniently propagates this added information through a straightforward extension of the propagation algorithm of Thwaites et al. (2008). Of course, the discriminatory power of the temporal information is dependent on how different the competing holding time distributions are.

5.7 Application of the Dynamic Falls Intervention

We now revisit the falls intervention described in Chapter 4. We extend this intervention to a dynamic setting where we consider the effects of the intervention on those who have been assessed in the community and in communal establishments. We shall group together high risk individuals who are treated with and without being referred. We shall also assume here that low risk individuals do not receive any treatments under the longitudinal intervention.

Further, we assume that for high risk individuals who suffer a fall there are three possible outcomes: 1) serious complications or death resulting from the fall due to which they leave our population under consideration, 2) no serious consequences and can resume normal life, and 3) those living in the community may be moved to a communal establishment for further care and support. For a low risk individual, we consider that a fall triggers

a reevaluation of their risk status. Figure 5.14 shows the event tree describing the process for assessed individuals in the community where the three dots following a vertex indicate that the tree continues on. Vertices s_{12} and s_{15} represent that the individual in the community might fall again while continuing to remain in the community. Hence, the possible consequences of a potential second fall are the same as earlier: "Return to normal", "Move to communal establishment" and "Complications". Whereas, vertices s_{13} and s_{16} represent a move to a communal establishment. Hence, the possible consequences from here of a potential second fall are "Return to normal" and "Complications".

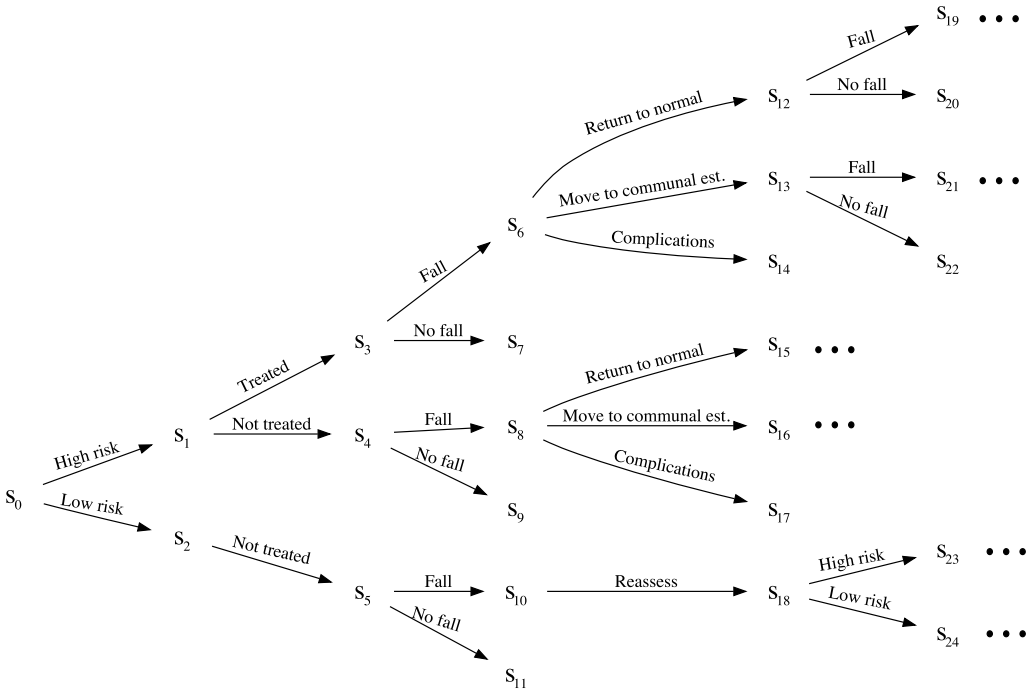


Figure 5.14: Event tree describing the dynamic falls intervention for assessed individuals in the community.

We simulated data for this longitudinal extension of the falls intervention by calibrating these to summary statistics provided in various studies, as described in Section 4.5. We test the performance of the model selection algorithm in the special case described in Section 5.3.3. Hence, we will assume here that the transition probabilities and holding times remain invariant to the number of times the individual has experienced a fall. This is a strong assumption and we emphasise here again that this is for simplicity. We discuss more promising potential model selection algorithms for the CT-DCEG class in Section 5.8.

With the known ground truth, we first analyse the performance of our model selec-

tion algorithm on the special case described in Section 5.3.3. Recall that model selection involves identifying the collection of stages in the underlying tree. However, unlike in a CEG, stages in a CT-DCEG must satisfy two conditions: one associated with the conditional transition distribution and the other with the conditional holding time distributions. Since the transition and holding time distributions are mutually independent, we can split the task of identifying the stages into two sub-tasks: identifying the sets of situations that satisfy the condition associated with the conditional transition distributions, and identifying the sets of edges that satisfy the condition associated with the conditional holding time distributions. Call these sets of situations as *situation clusters*, and the sets of edges as *edge clusters*. This, in fact, gives us another way of visualising staged trees and CT-DCEGs where vertex colours represent situation clusters and edge colours represent edge clusters. Of course, the set of positions – i.e. the vertex set of the CT-DCEG not including the sink vertex – would remain unchanged. In cases where most situations satisfy only one of the conditions of a stage, the added information obtained from the graph of a CT-DCEG from such an alternative visualisation might prove useful.

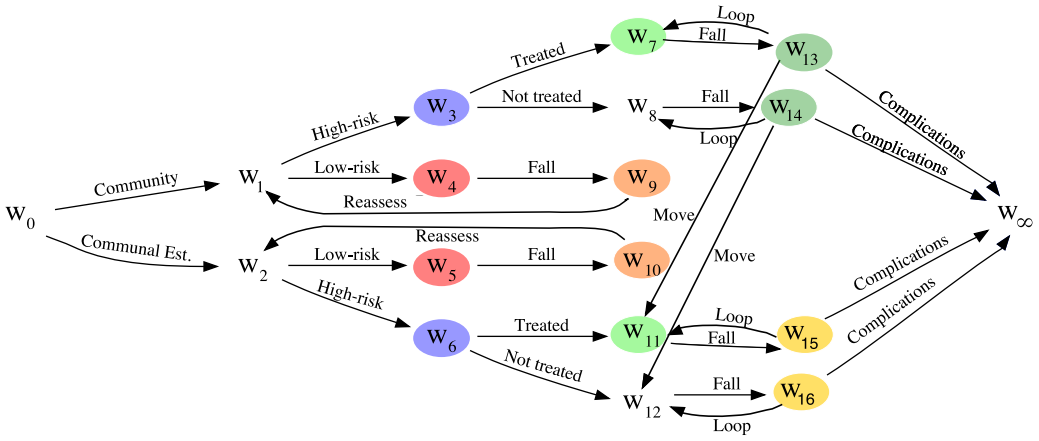
Random variable	**Description**
$H(e_{3,6}), H(e_{12,19})$	Duration from treatment or recovery to fall for community high risk individuals who are treated.
$H(e_{4,8})$	Duration from assessment or recovery to fall for community high risk individuals who are not untreated.
$H(e_{13,21})$	Duration from treatment, recovery or move from community to fall for communal high risk individuals who are treated.
$H(e_{5,10})$	Duration from assessment to fall for community low risk individuals.
$H(e_{6,12}), H(e_{8,15})$	Duration from a fall to return to normal.
$H(e_{6,13}), H(e_{8,16})$	Duration from a fall to moving to a communal establishment.
$H(e_{6,14}), H(e_{8,17})$	Duration from a fall to leaving the population of interest due to complications.
$H(e_{10,18})$	Duration from fall to reassessment for community low risk individuals.

Table 5.4: $H(e_{i,j})$ refers to the holding time along edge e_{ij} from situations s_i to s_j in Figure 5.14.

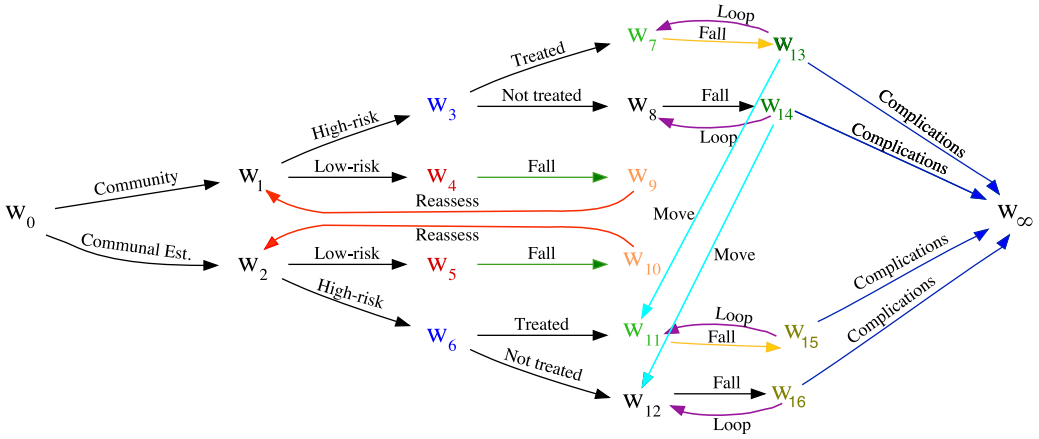
We analyse the data for simulations of sample sizes 500, 1500, 2500, 5000, 7500 and 10000. We simulated 100 instances of each population size. Edges emanating from vertices representing living arrangement (as community or communal establishment), level of risk (as high or low) and treatment status (as treated or not treated) are time-invariant and do not have any explicit holding time distributions. Transitions between states have Multinomial distributions and the conditional holding times, where appropriate, are generated

from two-parameter Weibull distributions. Table 5.4 describes the conditional holding time variables for the event tree segment shown in Figure 5.14. Figure 5.15(a) shows the graph of the data generating CT-DCEG for our longitudinal extension of the falls intervention, whereas Figure 5.15(b) shows the alternative visualisation described above. In both CT-DCEG graphs, the edge representing "No fall" emanating from vertices $w_4, w_5, w_7, w_8, w_{11}$ and w_{12} are hidden to prevent visual cluttering. The edge labels "Loop" and "Move" correspond to "Return to normal" and "Move to communal establishment" respectively. We can see in Figure 5.15(b) that the alternative visualisation can provide additional information that the original visualisation may not be able to show. In this case, we can see from Figure 5.15(b) that the distribution governing the time it takes from a fall to returning to normal life or to deteriorating with severe complications are conditionally independent of the individual's living arrangement given that they are classified as high risk. In fact, we can see that the time to returning to normal life given that an individual has fallen follows the same distribution, irrespective of their living arrangement and whether they received any treatment. Further, while the probability of falling is conditionally independent of the individual's living arrangement given that they are treated, this relationship does not hold when we condition on the individual not being treated. This is a form of contextual conditional independence that CEGs, including their dynamic variants, are able to express directly through their graph topology. Finally, notice that the dynamic falls intervention has structural zeros as those who are low risk do not get treatment with probability one and hence, there is no information lost by not representing vertices associated with the treatment variable for low risk individuals.

For identifying the situation clusters and edge clusters, we use the AHC algorithm across a wide range of prior specifications. Similar to hyperstages defined in Section 4.4, hyperclusters can be defined for both the situation clusters and the edge clusters. We analyse the situation clusters over imaginary sample sizes in the range of 0 to 100 in increases of 0.5, and the edge clusters over pseudo-holding times in the range of 1 to 200 in increases of 1 and for a fixed imaginary sample size of four. Because of the conjugacy properties of this class the search can be evaluated very quickly. We were able to retrieve the correct number of situation clusters (11) in most cases, particularly for the larger population sizes (5000 and above) for a wide range of prior specifications, see Figure 5.16(a). The edge clusters, however, typically returned more clusters (10 or 11) than we had in the generating model (9), see Figure 5.16(b). On closer inspection, we found that they are almost entirely caused by the AHC algorithm not being able to correctly identify the edge cluster made up of the edges representing a complication following a fall for high risk individuals (treated and not treated). In other words, this edge cluster contains the edges $e_{13,\infty}, e_{14,\infty}, e_{15,\infty}$ and $e_{16,\infty}$. The likely cause for this is that in our generating model, "Complications" are the least likely



(a)



(b)

Figure 5.15: (a) The data generating CT-DCEG for the simulated dynamic falls intervention; (b) an alternative visualisation of the same CT-DCEG.

outcome for a high risk individual who has fallen. Hence, even in large population sizes, they have relatively the least number of observations within our dataset. However, we would still expect these to be correctly identified in larger population sizes. One hypothesis is that this may be caused by the shape parameter of the generating Weibull distribution being less than 1. All the other generating Weibull distributions have a shape parameter greater than or equal to one and were always correctly identified. While this needs further investigation, we note that this does not appear to be a serious misidentification and the resultant CT-DCEG obtained by using the AHC algorithm to identify the edge clusters would still be close to the generating model in terms of its parameters as we shall see below.

The accuracy and stability of the average number of the situation and edge clusters, across the simulations for varying priors, typically improve as we observe more data. The AHC algorithm generally becomes more accurate and more discriminating as it receives more information. However, as with BNs (see e.g. Silander et al. (2007)), while several of the MAP models found have minor structural differences (in the number and composition of stages when compared with the generating model), they are similar in terms of the inference we can draw from them. While several measures could be used for analysing the robustness of the AHC algorithm, here we demonstrate this by calculating the Kullback-Leibler (KL) divergence as described below.

Let $\mathcal{D}$ be the generating CT-DCEG and $\mathcal{D}'$ be the model found by the AHC algorithm. Let S be the set of situations and E^* be the set of edges with conditional holding time distributions in its underlying invariant subtree. The KL divergence for the situation clusters $D_{KL}(S)$ is given as follows

$$D_{KL}(S) = \sum_{s_i \in S} \sum_{j=1}^{k_i} \theta_{ij} \log\left(\frac{\theta_{ij}}{\mathbb{E}[\hat{\theta}_{ij}]}\right) \tag{5.23}$$

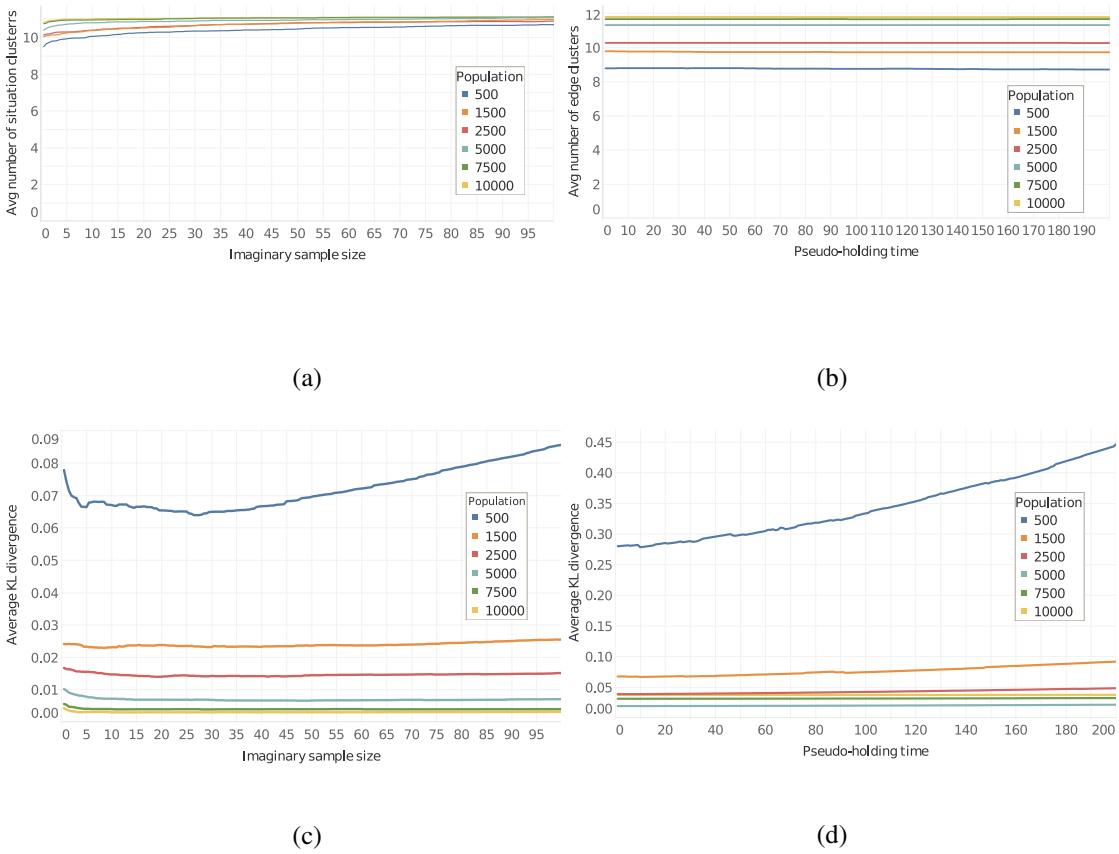
where vertex s_i has k_i emanating edges, θ_{ij} is the underlying true conditional transition probability of traversing the jth edge emanating from vertex s_i and $\mathbb{E}[\hat{\theta}_{ij}]$ is its corresponding mean posterior probability. Similarly, we define the KL divergence for the Weibull edge clusters $D_{KL}(E^*)$ as outlined in Bauckhage (2013) and given below

$$D_{KL}(E^*) = \sum_{e_{ij} \in E^*} \left\{ \kappa_{ij} \log\left(\frac{\mathbb{E}[\hat{\pi}_{ij}]}{\pi_{ij}}\right) + \left(\frac{\pi_{ij}}{\mathbb{E}[\hat{\pi}_{ij}]}\right)^{\kappa_{ij}} - 1 \right\} \tag{5.24}$$

where κ_{ij} and π_{ij} are the known shape parameter and unknown scale parameter of the generating Weibull distribution for edge e_{ij}, and $\mathbb{E}[\hat{\pi}_{ij}]$ is the expectation of the corresponding Inverse-Gamma distribution for the unknown scale parameter.

From Figures 5.16(c) and 5.16(d), we can see that the probabilistic accuracy of the

models improves very quickly in response to moderate increases in the population size. The KL divergence for the situation and edge clusters are relatively higher and more subject to volatility for the population size of 500. However, for population sizes of 1500 and more, there is increased stability across prior specifications.



(a) (b)

(c) (d)

Figure 5.16: The average number of situation clusters (a) and the corresponding average KL divergence (c) for varying values of the imaginary sample size. The average number of edge clusters (b) and the corresponding average KL divergence (d) for varying values of the pseudo-holding time and a fixed imaginary sample size equal to four.

In heterogeneous populations, information is often skewed against the smaller vulnerable groups. Here, we set the generating model such that there is disparity between the observations of high risk and low risk individuals. In spite of this, the CT-DCEG framework performs well for both risk groups. While the CT-DCEG model class used here assumes re-

current falls to be mutually independent due to our modelling choice for simplicity, experts might argue that an abnormally large number of falls could be indicative of the individual suffering from a chronic condition like Parkinsonism. In the CT-DCEG framework it is easy to embellish the model to incorporate such new hypotheses simply by adding new well-defined positions. Indeed such embellishments using expert judgements would be encouraged through discussion aided by the graphical interface.

5.8 Conclusion

In this chapter we introduced the class of continuous time dynamic CEGs. We vastly extended the preliminary work on extended DCEGs presented in Barclay et al. (2015), which turns out to be a special subclass of our more general CT-DCEG class. Further, we demonstrated a simple case of model selection in this class and proposed a dynamic propagation scheme for CT-DCEGs inspired by a similar scheme for DBNs put forth by Kjærulff (1992). This propagation scheme rests on three developments we presented in this chapter. The first is the development of the necessary semantics to describe the analogue of time-slices for this continuous time setting which enable us to "unroll" the CT-DCEG. The second is the alternative representation to a CT-DCEG provided by a, possibly approximate, semi-Markov process. This development makes forecasting possible within our dynamic propagation scheme. The third and most important development is the extension of the CEG propagation algorithm in Thwaites et al. (2008) so that it can handle temporal evidence concerning holding times at vertices. The CT-DCEG is a large and powerful class of models and there are various promising avenues for future research following from the work presented in this chapter; some of which are described below.

The model selection special case presented in this chapter rests on a rather strong assumption but one that simplifies the model selection process and allows us to begin a conversation on model selection for CT-DCEGs. The CT-DCEG is a very rich class of models and our model selection assumption here restricts us to a limited subclass. However, note that the other methodologies described in this chapter are still valid to all CT-DCEGs. Freeman and Smith (2011b) proposed a model selection procedure for a dynamic variant of CEGs where the underlying event tree is not infinite but instead has changing stage structures at each discrete time point. The changing stage structure is described with an underlying steady model (Smith, 1979, 1981). With the terminology described in this chapter, this method is equivalent to the repeating subtrees being structurally isomorphic to the invariant subtree. However, the unrolled passage-slices $C_{P(k)}$, for $k \in \mathbb{N}$ would not necessarily be isomorphic to each other in the structure or colour preserving sense as the stage structure, and hence, the position structure of the underlying invariant/repeating subtrees could change for

each k. Thus, the method in Freeman and Smith (2011b) can be employed for model selection in CT-DCEGs using the invariant subtree as the required finite event tree. However, the key difference is that the staging structure would change at each consecutive passage-slice rather than at each discrete time step. Another possible model selection methodology for CT-DCEGs could be based on non-stationary DBNs (J. W. Robinson & Hartemink, 2008; Grzegorczyk & Husmeier, 2009) where the structure and/or parameters are allowed to vary among time-slices based on Bayesian multiple change-point process. However, care must be taken while implementing such models to not overfit the time series data under the increased flexibility of these models. The suitability and applicability of such methodologies with respect to the CT-DCEG model class would need to be further investigated.

Further, notice that unrolled graph of a CT-DCEG generalises its original graph in the sense that it offers the flexibility to set arbitrary distributions for the parameters in each passage-slice. If the interest is in only a small number of passage-slices, the stage structure for the unrolled CEG over the corresponding passage-slices can directly be learned using the AHC algorithm on its situations and edges to identify the situation and edge clusters respectively. As before, we can use the log marginal likelihood score given in Equation 5.6 for this purpose, and conjugate updating can be performed as described in Section 5.3.1.

In our propagation algorithm, we restricted the elements of our temporal evidence to be point observations. In continuous time settings, it is often of interest to be able to propagate information obtained through interval temporal observations, e.g. the individual was in the state represented by vertex w_i from time $[t_1, t_2)$. Within CTBNs, such interval observations have been propagated using expectation-propagation algorithm as described in Nodelman et al. (2005) and Saria et al. (2007). This method could be further explored for CT-DCEGs.

Another interesting extension of our methodology would be to incorporate additional heterogeneities introduced by interdependence of adverse repeated events (such as repeated falls in our dynamic falls example). This would particularly be beneficial in the domains of reliability, survival analysis, ecology and conservation studies. The motivation for such developments can be drawn from conditional frailty models used in survival and event history analyses (Clayton, 1991; Box-Steffensmeier & De Boef, 2005; Box-Steffensmeier et al., 2007). These models incorporate heterogeneities across individuals as well as those induced by event dependence using random effects. The key idea behind this is that while a group of individuals may observe the same sequence of events (given by a walk in a CT-DCEG) and may have similar patterns in experiencing these events, the correlations and timings in the occurrences of future events for each individual (even if the same future events are observed by each individual) may be dependent on the correlations and timing of the occurrence of their past events.

We note that the developments presented in this chapter immediately allow us to develop a special subclass of discrete time DCEGs where edges may have arbitrary discrete holding time distributions. A model from this subclass would have an alternative representation in the form of a discrete time semi-Markov process. Time-invariant covariates could be incorporated in this special subclass in the same way as described in this chapter.

An interesting subclass of CT-DCEGs is explored in the next chapter. It is called the reduced dynamic chain event graph (RDCEG) (Shenvi & Smith, 2019) and it is based on conditioning on individuals not dropping out of our population of interest. This results in the RDCEG being valid for a different population as long as it shares the same missingness mechanism as the population for which it was developed. Note here, that this is countercultural to how missingness is typically handled in public health, as it redefines the target population (for modelling of informative dropouts in medicine, see e.g. Billingham and Abrams (2002) and Alaa et al. (2017)). We explore the benefits and drawbacks of the RDCEG in the next chapter when we review its application to a policing process as presented first in Smith and Shenvi (2018) and later extended in Bunnin and Smith (2019).

Chapter 6

Bayesian Modelling of Criminal Collaborations with CEGs

In this chapter, we demonstrate how CEGs can be used in combination with other models to describe a complex longitudinal process, where each model is a component within a larger composite model representing a distinct aspect of the process. We present here a policing application to model criminal collaborations between individuals within a population of suspects. This application is a culmination of a series of developments which are described below and presented in detail later in this chapter.

The use of CEGs for modelling criminal and policing applications was initiated in Smith and Shenvi (2018). This research report introduced a new subclass of continuous time DCEGs called the *reduced dynamic chain event graph* (RDCEG) and presented some generic features of an RDCEG for modelling lone criminals based on empirical sociological and psychological analyses of criminals involved in assault and violent crimes. The RDCEG subclass involves conditioning on individuals not dropping out of the population of interest. This model subclass was formally developed in Shenvi and Smith (2019) which also presented two public health applications of this novel subclass: the first, a public health intervention to reduce falls-related injuries among the elderly and the second, an intervention to assess the effects of delayed treatment by anti-epileptic drugs on those who present with early epilepsy and single seizures. Extending the application framework presented in Smith and Shenvi (2018) and the model foundations in Shenvi and Smith (2019), Bunnin and Smith (2019) then presented a three-level hierarchical model, called the *radicalisation and violent extremism* (RVE) model – with an RDCEG at its deepest level – to support the police in monitoring the progression of an individual to a violent attack against the general public. The criminal collaboration model for a population of suspects presented in this chapter builds on this existing work, and has two main components: a collection of RVE

models, one for each suspect in our population, and a novel dynamic network model.

Based on the series of developments presented above, this chapter is organised as follows. Section 6.1 motivates the modelling of lone criminals and criminals acting in concert with the three-level RVE model and the two-part criminal collaboration model respectively. Section 6.2 introduces the RDCEG model subclass, and discusses its scope and limitations. Most importantly, we discuss what this model subclass' conditioning on not dropping out implies about the missingness mechanism of the process being modelled, and how it redefines the target population. In Section 6.3 we present a review of the RVE model for modelling the progression of lone criminals to violent attacks against the general public. This review is largely based on the work of Bunnin and Smith (2019), with examples drawn from Smith and Shenvi (2018). Section 6.4 briefly reviews the existing literature of statistical network models for analysing criminal networks. In Section 6.5 we present a novel dynamic network model that can integrate pairwise observational communications data with prior knowledge on suspected individuals to determine the extent of information directly exchanged between pairs of suspects. By assuming certain plausible conditional independencies, we show how the distributions of the random variables measuring the pairwise information exchange can be updated in closed form using the concept of steady modelling (Smith, 1979, 1981). Further, we demonstrate in Section 6.6 how this network model can be integrated with the individual hierarchical RVE models to construct an integrating decision support system (IDSS) (Leonelli & Smith, 2015). An IDSS refers to a network structure connecting different models that integrate expert knowledge and judgement to enable reasoning about distinct aspects of a complex system into a coherent and consistent tool which could be used for decision making. To construct such an IDSS, here we use a decoupling methodology first introduced for the class of Multiregression Dynamic Models (MDMs) (Queen & Smith, 1993). We show how this methodology can be easily transferred to our setting. Finally, we use this IDSS to construct simple yet informative threat scores for a known or suspected criminal cell that would enable policing authorities to monitor the evolution of the threat they pose as a collaborative unit. In Section 6.7 we review a practical application of our criminal collaboration model using a simple example with simulated data. Finally, in Section 6.8 we conclude with a discussion.

The criminal collaboration model presented in this chapter derives from a collaboration with F. Oliver Bunnin and Jim Q. Smith under the Alan Turing Institute's Defence and Security programme, reported in the pre-print Bunnin et al. (2020). Section 6.7 is the work of Bunnin and is presented here, with permission, for completeness. Sections 6.4, 6.5.1, 6.5.2 and 6.6.3 were jointly developed with the co-authors.

6.1 Motivation and Introduction

Individuals who plan to commit acts of violence against the general public typically need to carry out a sequence of preparatory tasks before they can execute their plans. Policing authorities, after acquiring the necessary legal permissions, can monitor the activities of suspected criminals in order to protect public safety (see e.g. "Investigatory Powers Act (c.25)" (2016)). While these suspects may consciously try to hide or disguise their behaviour, their engagement in the necessary preparatory tasks will typically give rise to some observable data. This data may very well be incomplete and noisy. However, when combined with domain expertise, they can be informative of the progression of a suspect to an attack. Bunnin and Smith (2019) proposed a three-level hierarchical model, hereafter called the *Radicalisation and Violent Extremism* (RVE) model, for extracting signals from noisy observable streaming data on suspected criminals to provide the policing authorities with a tool to evaluate the imminence of the threat posed by a suspect acting as a lone criminal.

At the deepest level of this hierarchical model is a reduced dynamic chain event graph (RDCEG) model which was first introduced in Smith and Shenvi (2018) and formally developed in Shenvi and Smith (2019). The defining property of the RDCEG is that it conditions on individuals – within an open population – not dropping out of the population of interest. The open population implies that individuals can enter and leave the model from almost any state. The RDCEG focuses inference on those *who continue to be part* of the population of interest. Hence, the RDCEG is useful within domains where missingness is a common feature yet the missingness mechanisms (i.e. the mechanism describing how the missing values are distributed within the data) are hard, if not impossible, to identify with any amount of certainty. Policing is one such domain.

Further, individuals intent on terrorism may act in concert with other like-minded people to co-ordinate themselves so as to present a much more severe threat (Anderson, 2016; Kirk-Wade & Allen, 2020). The structure, scope, dynamics and intent of such criminal networks can be very diverse (Morselli, 2009). Use of statistical techniques from social network analysis (SNA) to analyse criminal networks is a very active area of research (see e.g. Sparrow (1991), Krebs (2002), Berlusconi et al. (2016), Broccatelli et al. (2016), and D. Robinson and Scogings (2018)). While identification and disruption of the activities of criminal networks is a prime objective of police and security forces (Europol, 2018; Kirk-Wade & Allen, 2020), this comes with challenges similar to those faced when modelling lone criminals: criminals may hide or disguise their intentions and activities; personal communications are private and are often encrypted, and numbers and powers of the policing authorities are rightly limited in democratic societies. However, criminals *do* need to perform certain activities and to communicate in order to organise and execute attacks. Just as

in the case of lone criminals, activities and communications within a criminal collaboration typically give rise to some observable data that in conjunction with domain knowledge can be used to construct statistical models to aid prevention and disruption of attacks. Again, policing authorities can observe these suspected individuals after the necessary legal permissions are sought.

Performing a multivariate extension of the RVE modelling technologies to model criminal networks in a way that it takes into account the connections between the individuals within a criminal group is far from straightforward. It requires bespoke graphically supported probabilistic models of groups of criminals over a given population cooperating in a way that leads to an attack cell. This type of model then needs to be combined in a coherent way with the RVE models we have of the individuals within that population so that the criminal threat posed by each member of the group can also be taken into account. In this chapter, we propose a new class of models that is able to do this. This model, called the *criminal collaboration model*, is described below.

Our criminal collaboration model, which was first reported in Bunnin et al. (2020), consists of two parts: (1) a weighted dynamic network model that analyses pairwise links among suspects forming the vertices of the network, and (2) the collection of individual RVE models of each suspect in the network. The latter are as described in Bunnin and Smith (2019) and reviewed in Section 6.3. The network model is composed of suspects within the criminal network as vertices. An edge between two individuals indicates a potential *collaborative link* between them. The weight on the edge measures the extent to which information is being shared between two individuals. Observations of pairwise communications between individuals are then used within a *steady model* (Smith, 1979, 1981) formulation to estimate the random variables modelling these edge weights. Through the steady model, our network takes into account the temporal dimension of the evolving link between suspects. Such links between individuals are informative of the flow of information between them and hence, may inform potential joint attacks. Additionally, the steady model formulation enables our inferences to be driven by closed form recurrences. This not only means that the method computes forecasts quickly in real time and its inference is scalable to much larger networks, but also that the model and its parameters remain transparent and interpretable throughout. We emphasise that the latter property is essential in sensitive domains such as policing, and empowers the decision makers to make well-informed and defensible decisions guided by the model. This guided our decision to use a simple bespoke dynamic network methodology with steady evolutions rather than using alternative more established network methods (Goldenberg et al., 2009; Fortunato & Hric, 2016) such as stochastic block models (e.g. Airoldi et al. (2008) and Xing et al. (2010)) and latent space network models (e.g. Hoff et al. (2002) and Sewell and Chen (2016, 2017)) which typically

result in the need for MCMC or variational methods to deal with the loss of conjugacy.

To operationalise the criminal collaboration model, it is necessary to integrate the individual RVEs with the dynamic network model. We can seamlessly combine the outputs of the RVEs and the dynamic network model by using the decoupling methodology of MDMs (Queen & Smith, 1993). This effectively makes our criminal collaboration model an IDSS (Leonelli, 2015; Leonelli & Smith, 2015; Smith et al., 2015) while retaining the closed form recurrences and hence, transparency across the decision support system. We then demonstrate how we can use this integrating system to define suitable *cell-level threat scores* to estimate the current and future threat posed by a known or suspected criminal cell. Thus we present a network methodology that is customised to the prevention of criminal attacks through the use of domain knowledge and data available to the counter-terrorism and policing authorities. Further, this system is flexible enough to implement interventions to the system in real time. The class of models we propose here is to our knowledge entirely new.

The work presented in this chapter is part of a larger initiative at the Alan Turing Institute to develop a decision theoretic framework to model potential individual and group criminal attacks as well as the action space and objectives of policing authorities. Thus, modelling the following issues comes under the purview of the project:

1. Progression of an individual to attack;
2. Social network among potential criminals;
3. Progression of a known or suspected group to an attack;
4. Identification of new potential groups.

The first of these was addressed by Smith and Shenvi (2018) and Bunnin and Smith (2019). The work presented in this chapter, reported in Bunnin et al. (2020), aims to address the second and third points. We discuss approaches to address the final point in Section 6.8.

6.2 The Reduced Dynamic Chain Event Graph

In this section, we introduce the RDCEG subclass of the CT-DCEG class introduced in Chapter 5. The RDCEG subclass was specifically developed to deal with challenges associated with modelling processes based on open populations. Depictions of open populations, i.e. populations where people can immigrate and emigrate, occur widely in ecology, conservation and epidemiology, see e.g. Goffman (1965) and Nisbet and Gurney (1982). The individuals of these populations are in a constant state of flux due to a variety of reasons. This might include individuals moving from the region of study, their failing or improving health or death.

In the contexts described in this chapter, although emigration causes missingness, the processes driving the missingness are not only difficult to correctly identify (Little et al., 2017) but are often not random in any sense – i.e. under the classification introduced in Rubin (1976), the missingness mechanism is typically missing not at random (see Section 6.2.2). In several scenarios, such as the policing application discussed later in this chapter, we find that the population on which inference usually needs to be made is *within* the dynamically changing extant population and not the larger general population from which this subpopulation is selected. An example of such a scenario is given below.

Example 6.1 (Smoking intervention)**.** *Suppose that a public health body wishes to analyse the effectiveness of its smoking cessation programme among those who register and stick with the programme. In this case, the inference we would be interested in conducting is not on the general population of smokers within the catchment area of the public health body, but rather on the subpopulation of individuals – drawn from this population – who register for the smoking cessation programme and stick with it. Of course, the public health body may very well be interested in understanding what drives registration and retention within the programme. However, these are distinct from the modelling purpose described above.*

Later in this chapter, we look at a policing example where an RDCEG is involved in modelling the activities of suspected criminals. In this case, we note that individuals may leave our population of interest for reasons such as being arrested, deported, dying or perhaps even choosing to leave behind a life of crime. The local policing authorities typically are concerned with the possibility of criminal acts of violence against the general public within their jurisdiction. Hence, once an individual drops out of the subpopulation of threatening criminals within the jurisdiction of the local authorities, they may no longer be of interest to them. Thus, given already existing shortages of personnel and finances, the authorities may only be interested in those individuals who *continue to* pose a risk. The graphical model used to represent this process must reflect this. We note that there are likely to be regional, national and international security bodies who may also be interested in monitoring those individuals who used to pose a threat in the past but don't anymore. Again, as far as the local policing authorities are concerned, that is not within their purview and modelling the currently extant subpopulation is sufficient for their purposes. However, great care must be exercised to not generalise the results of analyses carried out on this subpopulation, see discussion in Section 6.2.2.

For the reasons described above, within such a setting we find it expedient to build statistical models of the extant population directly rather than using the non-ignorable response methods which would necessarily involve an additional model of the, here very complex, missingness mechanism (Little & Rubin, 2019). Hence, the RDCEG is parametrised

conditioning on individuals *not dropping out*. We further discuss the implications of such conditioning on missingness in Section 6.2.2.

6.2.1 Model Description

Consider an infinite event tree $\mathcal{T}$ with vertex set $V(\mathcal{T})$ and directed edge set $E(\mathcal{T})$. We say that an event represented by label l is a *terminating event* if its associated edge $e = (v, v', l)$ is directed into a leaf in $\mathcal{T}$, i.e. $v' \in L(\mathcal{T})$. Suppose that the event tree depicts a process on an open population. Our target subpopulation is those who continue to engage in the process being modelled. Individuals can leave this subpopulation of interest from any state for a variety of reasons that are not directly associated with the primary purpose of modelling. The events associated with these reasons can be grouped under an umbrella event with the label "dropping out". Notice that the event of "dropping out" is necessarily a terminating event. Moreover, there might be other terminating events, which are within the scope of the primary purpose of modelling, that individuals in the population could experience. Say that these events are *critical terminating events*.

Example 6.2 (Smoking intervention continued)**.** *In this example, the public health body will be interested in studying individuals who have left our subpopulation of interest because they have quit smoking. Whereas, given the primary purpose of their study, they may be less interested in those who leave the population for reasons such as moving out of the catchment area, dropping out due to peer pressure, being unable to continue the programme due to hospitalisation or poor health. In this case, the terminating event of "quitting smoking" can be classed as a critical terminating event while the other reasons for leaving the population may be grouped under the non-critical terminating event "dropping out".*

As the RDCEG depicts events relevant to the extant population, we depict the critical terminating events explicitly within its graph. The choice of which terminating events are considered as critical depends on the application and the purpose of modelling. Let $C(\mathcal{T}) \subseteq L(\mathcal{T})$ be the set of leaves into which edges associated with critical terminating events enter, and $D(\mathcal{T}) = L(\mathcal{T}) \backslash C(\mathcal{T})$ be the set of leaves associated with non-critical terminating events, i.e. dropping out for other reasons.

Definition 6.3 (Modified Event Tree)**.** *The modified event tree $\mathcal{M}$ of an event tree $\mathcal{T} = (V(\mathcal{T}), E(\mathcal{T}))$, with $D(\mathcal{T})$ denoting its set of leaves associated with non-critical terminating events, is obtained as the subgraph of $\mathcal{T}$ induced by the vertices $V(\mathcal{T}) \backslash D(\mathcal{T})$.*

The transition probabilities along the edges emanating from each situation of $\mathcal{M}$ are renormalised to ensure that these probabilities sum to one for every situation. Denote the transition probability parameters for the modified event tree $\mathcal{M}$ by $\Phi_{\mathcal{M}} = \{\boldsymbol{\theta}^*_v | v \in S(\mathcal{M})\}$

where $\boldsymbol{\theta}_v^*$ is the renormalised probability vector for situation v. Denote by $\boldsymbol{\mathcal{H}}_{\mathcal{M}} = \{\mathbf{H}(v)|v \in S(\mathcal{M})\}$ where $\mathbf{H}(v) = \{H(e)|e = (v, v', l) \in E(\mathcal{M}), v' \in \text{ch(v)}\}$ is the vector of conditional holding time random variables associated with the edges emanating from situation v in $\mathcal{M}$. Note that the holding time distributions along the edges in $\mathcal{M}$ are inherited directly from $\mathcal{T}$.

The process of constructing an RDCEG from its modified event tree is identical to the process of constructing a CT-DCEG from its underlying event tree. Denote by $\mathbb{U}^*$, $\mathcal{S}^*$ and $\mathbb{W}^*$ the collection of stage sets in $\mathcal{M}$, the staged tree obtained by colouring vertices of $\mathcal{M}$ according to their stage memberships, and the collection of position sets in $\mathcal{M}$ respectively. Denote by $\Phi_{\mathcal{S}^*}$ the conditional transition parameters in $\mathcal{S}^*$, and by $\boldsymbol{\mathcal{H}}_{\mathcal{S}^*}$ the conditional holding time variables in $\mathcal{S}^*$.

Definition 6.4 (Reduced Dynamic Chain Event Graph)**.** *A reduced dynamic chain event graph (RDCEG) $\mathcal{D} = (V(\mathcal{D}), E(\mathcal{D}))$ is defined by the tuple $(\mathcal{S}^*, \mathbb{W}^*, \boldsymbol{\Phi}_{\mathcal{S}^*}, \boldsymbol{\mathcal{H}}_{\mathcal{S}^*})$ with the following properties:*

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

The main feature of the RDCEG is its conditioning on not dropping out of the population, and hence, redefining the population of interest or the target population. Note that the RDCEG can also be analogously defined as a subclass of the discrete time DCEG or any other dynamic variant of the CEG family as long as it retains this key property.

6.2.2 Implications on Missingness

We first note that there are three main categorisations of missingness defined by Rubin (1976) as given below

- Missing completely at random (MCAR) which implies that the missingness does not depend on the observed or unobserved values. Here the missing values are a random subset of the complete data.
- Missing at random (MAR) which implies that missingness depends on the observed values but not the unobserved values.

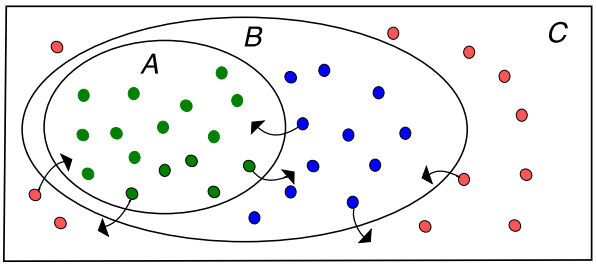


Figure 6.1: In this figure, C marks the whole underlying population, B marks the general subpopulation of C with the properties that define the individuals the process being studied applies to, and A marks the subpopulation of B that, given the opportunity, would choose to or be chosen to engage with the process.

- Missing not at random (MNAR) which implies that the missingness depends on both the observed and unobserved data.

The processes for which the RDCEG is considered as a model choice are typically those where we have observational data on an open population, and where missingnesss is likely to be MNAR. While there exist statistical tests to identify whether the missingness is MCAR (Little, 1988), identification of MAR and MNAR are complicated by the fact that the data needed to test for these are missing. Schafer and Graham (2002) state that "missingness is usually a nuisance, not the main focus of inquiry, but handling it in a principled manner raises conceptual difficulties and computational challenges". Thus when dealing with studies where the missingness is MNAR, making inference about the general underlying population is not typically feasible. However, there is a possible solution in the case where we are interested in making inference not about the general underlying population but specifically about the subpopulation that, given the opportunity, chooses to or is chosen to engage in the process. More precisely, we can make inference about this specific subpopulation by conditioning on the missingness mechanism, which is what the RDCEG model does. Here "engaging with the process" depends on the application. For example, in the case of the smoking intervention example, it refers to the continual participation in the smoking cessation programme, whereas in the policing application, it refers to the continual decision of those in charge to monitor a suspect.

Example 6.5 (Smoking intervention continued)**.** *In this example, the whole underlying population (population C) consists of all the individuals within the catchment area of the public health body irrespective of whether they smoke. The general population of interest (subpopulation B) for this intervention is the population of individuals within the catchment area who currently smoke. Conventionally, the interest typically lies in making inference*

about this subpopulation B. The individuals who, given the opportunity, would engage in the smoking cessation programme form subpopulation A. Individuals can move between these three population categories (see Figure 6.1). The RDCEG model is concerned with inference about subpopulation A.

Note that in public health and medicine including clinical trials, the target population is typically what we mark as subpopulation B in Figure 6.1. In this case, except when the missing data is MAR or MCAR, it is inappropriate to not include a model for the missingness mechanism within the model likelihood (Rubin, 1976; Pajouheshnia et al., 2020) and hence, inappropriate to use an RDCEG model for such settings where the interest lies in subpopulation B. Within such cases, by conditioning on not dropping out as we do in the RDCEG, a mismatch is created between our target population and the population for which we can reasonably make any inference. To see this, suppose first that we have data $Y = \{Y_1, Y_2, \ldots\}$ and an observation indicator variable $R = \{R_1, R_2, \ldots\}$ where $R_i = 1$ implies Y_i is observed, whereas $R_i = 0$ implies it is missing. From Rubin (1976), we know that under the selection factorisation the likelihood given data Y will factorise as follows

$$L_{\text{full}}(\theta, \phi \,|\, y_{(1)}, r) = L_{\text{ign}}(\theta \,|\, y_{(1)}) L_{\text{rest}}(\phi \,|\, y_{(1)}, r), \tag{6.1}$$

only when the missingness is MAR, where $y_{(1)}$ is the observed part of the realisation of Y, r is the realisation of the observation indicator, and θ and ϕ are the parameter vectors of the data model and missingness model respectively which are independent of each other. That is, the likelihood of the data model parameter vector θ does not involve a model for the missingness R. Here, we can directly work with $L_{\text{ign}}(\theta \,|\, y_{(1)})$ and this term is sometimes referred to as the "ignorable likelihood" (Little et al., 2017). When the missingness is MNAR, the above factorisation in Equation 6.1 is not valid and generally, a model for the missingness mechanism must be explicitly included.

There are two key precedents for conditioning on the missingness mechanism as we do with the RDCEG. The first is Little et al., 2017 which states that an alternative way of factorising the full likelihood is

$$L_{\text{full}}(\theta, \phi \,|\, y_{(1)}, r) = L_1(\theta \,|\, y_{(1)}, r) L_2(\phi \,|\, r), \tag{6.2}$$

which arises from a pattern-mixture factorisation of the joint probability distribution of Y and R (Little, 1993). Inference for the parameter vector θ in the $L_1(\cdot)$ term in Equation 6.2 cannot be typically generalised to the subpopulation B as it conditions on the missingness R. It must be noted, however, that Little et al. (2017) used Equation 6.2 to write $\theta = (\theta^{(0)}, \theta^{(1)})$ where $\theta^{(0)}$ is the subset of θ about which we cannot make any useful interpretation for their target population (subpopulation B) as the data required to do so is missing, whereas subset

$\theta^{(1)}$ can still be generalised to the target population under certain conditions. Little et al. (2017) describe the conditions under which inference for $\theta^{(1)}$ can be generalised to the target population. This is quite different to what we are proposing with the RDCEG as we define our target population to be subpopulation A instead of subpopulation B. However, in this case, inference for θ in the $L_1(\cdot)$ term in Equation 6.2 applies to subpopulation A.

The other key precedent is the approach taken by Geneletti and Dawid (2011) to identify the causal estimate of the *effect of the treatment on the treated* (ETT). In fact, their motivating examples (see Example 1: Training programme and Example 2: Invalid randomisation in Geneletti and Dawid (2011)) are rather similar in structure to our smoking intervention example and the policing application respectively. In the first there is a voluntary choice of who joins subpopulation A, and in the second, individuals are chosen to be considered in subpopulation A by someone who has the power to do so (in the policing application, this would be those in charge of prioritising and de-prioritising cases). Geneletti and Dawid (2011) state that within observational settings we observe two distinct effects:

- A *treatment effect* which refers to the power of the treatment to influence the outcome of interest;
- A *selection effect* which refers to the fact that these individuals choose to be or are chosen to be part of the observed subpopulation and so we are not observing random subsets of the population of interest (referring here to subpopulation B).

The ETT is then defined such that it quantifies the treatment effect for the specific subpopulation of individuals who choose to or are chosen to take the treatment. Note that within interventional settings, the ETT would involve a decision variable indicating whether the individual would choose to or be chosen to take the treatment, irrespective of whether they are actually assigned and given the treatment. Based on the situations for which the RDCEG is appropriate, these are likely to be observational settings. The RDCEG is thus similar in spirit to the ETT where our inference is for those who either choose to be part of subpopulation A (e.g. in the smoking intervention example) or those who are chosen to be part of subpopulation A (e.g. in the policing application).

As stated in Geneletti and Dawid (2011) for the ETT, one has to be very cautious in generalising the inference made from an RDCEG to a different population. For two populations with different compositions and attitudes, the relevant observational settings would most likely be different. This implies that the distributions of individuals who choose to be part of subpopulation A in these two populations would be different or the behaviour of those who choose those who are of subpopulation A would be different. Thus inferences from an RDCEG model designed for a specific subpopulation would not be informative of individuals, drawn from a different population, who cannot be considered as exchange-

able with those in the study population. Conditions under which and to what extent these inferences can be transferable need further consideration.

6.3 Review of the Radicalisation and Violent Extremism Model

In this section, we review the RVE hierarchical Bayesian model for criminal investigations presented in Bunnin and Smith (2019) as well as the role of the RDCEG within it. An RVE model is constructed by introducing a hierarchical extension to an RDCEG model – first presented for modelling criminal activities in Smith and Shenvi (2018) – for an attack by a lone criminal. The hierarchical structure built on top of an RDCEG model enables us to extract relevant signals from noisy streaming data which are then used to estimate the parameters of the RDCEG model. The three-level hierarchical RVE model aims to support the police in their pursuit of violent criminals acting alone to commit a crime against the general public. Denote by Ω the open population of persons of interest (POIs) at time t. Let $\Omega_t \subseteq \Omega_t^*$ be the subset of individuals that the authorities have decided to investigate and monitor at time t. The three levels of the RVE model for a suspected criminal $\omega \in \Omega$, are described below.

Deepest level: At this level, the RVE consists of an RDCEG model. The specific choice of states/vertices to be depicted within the graph of the RDCEG model depend on the type of criminal behaviour the individual might be engaged in, for example a murder plot, a vehicle attack etc. The states represented by the vertices of the chosen RDCEG and its underlying event tree should be such that they reflect the possible paths of progression for the modelled criminal behaviour. Denote by W_t the latent random variable indicating the state occupied by the suspect ω at time $t > 0$. The sample space of W_t is given by the vertices $\{w_0, w_1, \ldots, w_n\}$ of the RDCEG.

The event tree of the RDCEG model also includes a state associated with the event of "dropping out" (called a "neutral" state in Bunnin and Smith (2019)) as described in Section 6.2. However, for the reasons described earlier, this state is not part of the RDCEG model itself. Within the RVE context, the "dropping out" or "neutral" state represents that the individual no longer presents a threat to the general public within the jurisdiction of the policing authority. Any states associated with critically terminating events, however, will be part of the RDCEG. Further, Bunnin and Smith (2019) recommend keeping the number of states depicted within the vertices of the RDCEG as small as possible while retaining sufficient information to distinguish the relevant states, and testing the suitability of the chosen states using a clarity test (Howard, 1988). To pass the clarity test, in the hypothetical situation where a suspect is asked to place themselves within a particular state, they must be able to do so without ambiguity.

Example 6.6 (Murder plot)**.** *We look at the RDCEG borrowed from an example in Smith and Shenvi (2018) which illustrates a murder plot using a gun. The RDCEG for this plot includes states given as "can't shoot, no gun" (w_0), "can't shoot, has a gun" (w_1), "trained to shoot, no gun" (w_2), "trained to shoot, has a gun" (w_3) and "attempts murder" (w_4). The "neutral" state is not depicted within the RDCEG model. Figure 6.2 shows the graph of this RDCEG. The vertex colouring has been suppressed in this graph. The sample space of W_t for time $t > 0$ is given by $\{w_0, w_1, w_2, w_3, w_4\}$. The events are represented by the edge labels in the graph. Note that while "attempts murder" is not an absorbing state here, it is conceivable to imagine a scenario where such a state is absorbing and hence is associated with the critically terminating event of "locating and approaching target". This would then imply that the suspect can only attempt the murder once irrespective of the outcome of the attempt.*

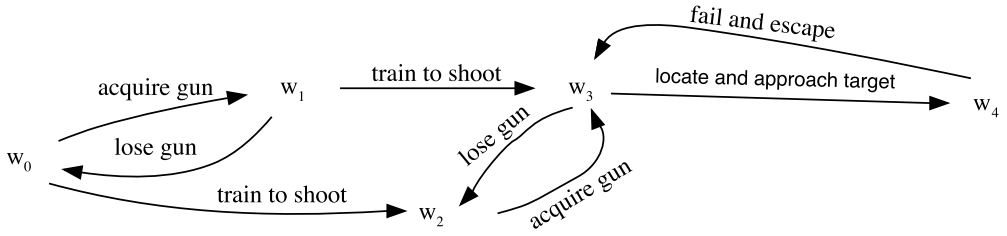


Figure 6.2: Graph of the RDCEG for the murder plot.

Of interest within the RDCEG model are the parameters of the transition probability and holding time distributions. In many instances within the domain of policing, routine measurements concerning suspects may be recorded and reported at fixed time intervals (Bunnin & Smith, 2019). In such cases, generally the process being modelled reduces to a discrete time process even though the underlying process evolves in continuous time (see Section 5.1). However, within this application, we find it beneficial to retain the underlying model as evolving in continuous time. This is discussed in Section 6.3.1.

Smith and Shenvi (2018) provide several types of categorisations for a wide range of criminal behaviours which can be used to inform the states of the RDCEG. The states of the RDCEG can also be customised to the individual ω to whom the RVE concerns such that they represent the history, personality, environment and modus operandi of ω. Lastly,

it is essential that the states of the RDCEG are defined such that a collection of tasks can be associated with each state. These tasks form part of the intermediate level described below.

Intermediate level: At this level, the RVE consists of a collection of R tasks associated with the RDCEG in the level below. At any time $t > 0$, denote the task vector at time t by $\boldsymbol{\theta}_t = \{\theta_{t1}, \theta_{t2}, \ldots, \theta_{tR}\}$ where each θ_{ti} is an indicator variable such that $\theta_{ti} = 1$ if ω is enacting task i at time t, for $i \in \{1, 2, \ldots, R\}$. Each task can be associated with one or more states of the RDCEG. The purpose of the task vector is to enable the policing authorities to estimate how far along the suspect is in their progression towards a specified criminal attack. Thus, the task vector at time t can not only provide positive evidence that the suspect ω is likely to be a certain state w_i but can also provide negative evidence that the suspect, initially believed to be in some position w_i, is not performing the tasks associated with this state and hence, must be in a different state.

Example 6.7 (Murder plot continued)**.** *Tasks for the example of the murder plot using a gun might include the following:*

$$\theta_{t1} = \textit{“Acquires a gun”}; \qquad \theta_{t2} = \textit{“Trains to shoot”};$$
$$\theta_{t3} = \textit{“Follows the target”}; \quad \theta_{t4} = \textit{“Secures monetary resources”}.$$

Here task $\theta_{t1} = 1$ for some time $t > 0$ provides positive evidence for states w_1 and w_3, and negative evidence for states w_0 and w_2. At the same time, if we had both $\theta_{t1} = 1$ and $\theta_{t2} = 1$ then the evidence points towards state w_3 rather than w_1. If we had $\theta_{t3} = 1$ and $\theta_{t4} = 1$ as well, this might provide evidence in favour of the suspect transitioning from state w_3 to state w_4.

The data informing ω's engagement in these various tasks may range from complete and reliable intelligence to partial and patchy secondary data. The amount of observations recorded about ω and the reliability of these observations are likely to depend on how closely the police are monitoring ω. The further away ω is from the main investigation, the more indirect the observations are going to be. However, even noisy signals obtained from partial data can help the police to condition on the limited information under the Bayesian RVE model and revise these judgements accordingly. The data on ω and the signals obtained from the data form a part of the surface level of the RVE.

Surface level: This level consists of the data $\{\mathbf{Y}_t\}_{t\geq 0}$ relating to activities of the suspect ω . This typically refers to the data collected by the policing authorities by directly monitoring the individual, when they have the necessary resources and authorisation to do so. However, it can also include secondary data that is typically received at irregular intervals. For each task θ_{ti}, we can associate a subset $Y_{ti} \subseteq \mathbf{Y}_t$ of the data stream observed at time t which informs whether ω is engaged in task θ_{ti}, for $i \in \{1, 2, \ldots, R\}$. It will

typically be necessary, especially if the data is noisy, to use a *filter* to obtain filtered data Z_{ti} from the data subset Y_{ti} to extract any viable signal from the noise. Denote by $\{\mathbf{Z}_t\} = \{Z_{t1}, Z_{t2}, \ldots, Z_{tR}\}$. Here a filter is simply a suitable function $\tau_i(\cdot)$ of the data Y_{ti}. It is essential that this filter is chosen such that it is sufficiently capable of identifying when the activities of the suspect indicate that they are actually engaging in a particular task rather than other innocent and benign activities.

Example 6.8 (Murder plot continued)**.** *Example of observations may include the following:*

$$Y_1 = \text{“Seen around target”}; \qquad Y_2 = \text{“Personal threats made to target”};$$
$$Y_3 = \text{“Visits to shops selling guns”}; \quad Y_4 = \text{“Meetings with trained radicals”}.$$

Here $\mathbf{Y}_t = \{Y_1, Y_2, Y_3, Y_4\}$. This data can be associated with the different tasks; for instance Y_3 can be associated with task θ_{t1}, in other words $Y_{t1} = \{Y_3\}$. An example from Bunnin and Smith (2019) of a filter τ_i for task θ_{ti} is given below

$$\begin{aligned} Z_{ti} &= \tau_i(\mathbf{Y}_t) \\ &= \frac{1}{|I(\theta_{ti})|} \sum_{i \in I(\theta_{ti})} \tilde{y}_{ti}, \end{aligned} \tag{6.3}$$

where $\tilde{y}_{ti}$ is the standardisation of Y_{ti} and $I(\theta_{ti})$ is the index set of the components of $\mathbf{Y}_t$ associated with task θ_{ti}. Here standardisation refers to subtracting a pre-defined mean from the observation and dividing it by a pre-defined standard deviation estimated by domain experts (Bunnin & Smith, 2019).

Other examples of simple filters include first-order differences, short term averages etc. of the derived data. Expert judgement and any available data would be essential here to distinguish the distributions of $p(Z_{ti} \mid \theta_{ti} = 1)$ and $p(Z_{ti} \mid \theta_{ti} = 0)$. The more distinct these distributions are, the easier it is to estimate whether $\theta_{ti} = 1$ in light of the filtered data Z_{ti}.

6.3.1 Recurrences in the RVE Model

For the RVE model to be operationalised within the domain of policing, it is essential that the initial state priors as well as the priors for the parameters of the transition probability and holding time distributions are set using expert judgement. Recall that, as described in Bunnin and Smith (2019), in many instances the observation data on individuals is sequential and recorded at regular discrete time points. However, the underlying evolving threat state of the suspect ω evolves in continuous time. Further, some of the data on ω may be recorded unexpectedly at irregular time intervals, as they arrive, due to the nature of the application. This typically is an issue with secondary data. Hence, for this type of application,

it is beneficial to still model the RVE model as evolving over continuous time despite many of the routine observations being recorded at fixed time intervals.

We briefly review here the recurrence equations for the RVE model as presented in the supplementary material of Bunnin and Smith (2019). The key point here is that within a semi-Markov process when the probability of transition over a particular interval of time is specified, its embedded Markov structure can be exploited to define the recurrence equations for the model.

We assume that within a short interval of time $(t, t']$ only one state transition can occur, for $t' > t \geq 0$. Let $H_i(t, t')$ denote the probability of transition out of state w_i during this time interval where H_i is the unconditional holding time distribution at state w_i. Note here that Bunnin and Smith (2019) use the unconditional holding time at a state w_i rather than the conditional holding times we use in Chapter 5 and Section 6.2. However, these recurrences can also be defined using conditional holding times, see e.g. Becker et al. (2000) and Moura and Droguett (2008). Further, denote by M^0 a matrix whose (i, j)th entry represents the probability of a transition from state w_i to state w_j. Denote by $M(t, t')$ the full transition matrix which is the product of the conditional transition probabilities and the state dependent probabilities of transition in the time interval $(t, t']$. The recurrence equations are then given as

$$p(W_t = w_i \,|\, \boldsymbol{Z}_{t-1}) = \sum_{w_j} p(W_{t-1} = w_j \,|\, \boldsymbol{Z}_{t-1}) M_{j,i}(t-1, t) \tag{6.4}$$

$$p(W_t = w_i \,|\, \boldsymbol{Z}_t) \propto p(W_t = w_i \,|\, \boldsymbol{Z}_{t-1}) \times \sum_{\theta \in \boldsymbol{\theta}_t(w_i)} p(\boldsymbol{Z}_t \,|\, \theta) p(\theta \,|\, W_t = w_i) \tag{6.5}$$

where $\boldsymbol{\theta}_t(w_i) \subseteq \boldsymbol{\theta}_t$ denotes the set of all tasks which are in any way relevant to state w_i.

The recurrences given in Equation 6.4 and 6.5 describe how we can combine the discrete sequential observational data with the underlying continuously evolving threat state on an individual. The detailed example in Section 5 of Bunnin and Smith (2019) models a suspect working towards a vehicle attack where the data updates occur on a weekly basis. Also in this example in Bunnin and Smith (2019), the "dropping out" or "neutral" state is included explicitly within the model and is only visually hidden in the graph.; i.e. there is no intermediate modified tree construction and the conditional transition probabilities are not renormalised after excluding the dropping out states and the edges entering them. The model in this example is therefore more akin to a CT-DCEG (connected to the discrete observations using the recurrences mentioned above) with the neutral state hidden in the graphical representation rather than an RDCEG as described in Section 6.2. However, the methodologies described in Bunnin and Smith (2019) are still applicable to, and in fact were designed for, an RDCEG model.

6.4 Related Research

In this section, we review the existing literature concerning statistical network methods used for analysing criminal networks. Network data relating to activities of opposition and criminal forces have been analysed using SNA methods including link analysis for a long time, going back to at least World War II (Departments of the Army and the Air Force, 1948; van Meter, 2002; Cunningham et al., 2015). Examples of link analysis and network survey methods being used to gain intelligence and strategise can also be found from The Troubles in Northern Ireland, and in Thailand in connection to the fight against the Communist rebels (van Meter, 2002).

Within academic contexts, the merits of network analysis for terrorism research were originally assessed by Sparrow (1991) in his seminal paper. The author motivated the importance of network analysis concepts such as centrality, node degree, betweenness, closeness, stochastic equivalence and Euclidean centrality after multidimensional scaling within the context of criminal and terrorist networks. He further emphasised issues such as "weak ties" which indicate that the most valuable and urgent communication channels are likely to be those "which are seldom used and which lie outside the relatively dense clique structures", "fuzzy boundaries" which indicate that boundaries of criminal networks can be quite ambiguous, and "incompleteness" indicating that data relating to criminal networks are likely to be incomplete with MNAR missingness. Prior to the work of Sparrow (1991), the leading method of network analysis within law enforcement was the Anacapa charting system (Harper & Harris, 1975) developed by Anacapa Sciences Inc., California and widely used since its introduction. This charting system provided a two-dimensional visual representation of the link data. However, Anacapa charts are primarily visualisation tools, allowing the user to clearly pick out features such as links, centrality, cliques etc but the charting system itself does not involve any analysis of the data these charts represent.

Following Sparrow (1991), the application of statistical network analysis methods within criminal and terrorist networks has been researched extensively. These include centrality measures to identify key individuals and heterogeneous roles (Lee et al., 2012; Toth et al., 2013), Bayesian bipartite graph methods to identify overlapping cells (Ranciati et al., 2020), multipartite graph methods to cluster similar terrorist groups (Campedelli et al., 2019), and dynamic line graphs to visualise the temporal dynamics of terrorist actors in covert actions and events (Broccatelli et al., 2016), and spectral clustering to identify criminal cells (van Gennip et al., 2013). A discussion on the use of link prediction methodologies in criminal networks can be found in Section 6.8.

The existing criminal network research, while being a growing field, is not always inclusive of the multitude of real-time regular and irregular data channels available to polic-

ing authorities. Often, the methodologies developed cater to a fixed data channel (see e.g. Iqbal et al. (2012), Ferrara et al. (2014), and Sarvari et al. (2014)). Criminal network research, while taking into account the connections and communications between the individuals, often fail to take into account the underlying trajectories of each individual within the network. Our criminal collaboration model aims to contribute to the field by proposing a novel statistical dynamic network, which takes into account the temporal evolution of the connections between pairs of individuals, *together with* structured stochastic processes describing the personal criminal trajectories of the individuals in the network.

6.5 The Dynamic Network Model

In this section we present the dynamic network model which along with the RVE model described in Section 6.3 forms our two part criminal collaboration model. The criminal suspects monitored by the policing authorities are the vertices in the dynamic network model, and an edge between two suspects indicates direct exchange of information between them. This dynamic network is weighted, and the weight associated with an edge between two suspects measures the extent of information being shared between them. We first describe in Section 6.5.1 the types of pairwise communications data that the policing authorities might have access to, which can be used to inform these edge weights. We then describe our dynamic network model in Sections 6.5.2 and 6.5.3.

6.5.1 Pairwise Communications Data

Collaboration between criminals gives rise to observable data of various kinds. In particular, individuals need to be able to communicate and exchange information between them in order to carry out a criminal act together. Within a criminal group, it is essential that each member of the group is able to share information with at least one other member of the group for the group to be functional as a joint unit. Therefore in our dynamic network model, we focus on measuring the *extent of information being directly shared between pairs of individuals*.

Policing authorities typically receive information from multiple data channels. Some example channels are the monitoring of physical meetings, interception of electronic communications, and intelligence obtained from other policing agencies, covert informants, or the public. There are at least five types of potentially knowable or observable data that can be obtained by the policing authorities:

- Existing kinship or social links;
- Work or other shared affiliations;

- Bilateral electronic communications (e.g. telephone, email, Whatsapp etc);
- Physical meetings (observed directly or through closed circuit television);
- Financial transactions (e.g. bank transfers between accounts).

The first two items are relatively static whereas the others are more dynamic. Moreover the first two are not necessarily caused by criminal collaboration, but may enable a pre-existing tie that facilitates collaboration once other factors have come into play. Examples of these ties are the school and social ties that existed between several of the Al-Qaeda September 11, 2001 terrorists (Krebs, 2002), the kinship tie between Saleem and Hashem Abedi, the former being the suicide bomber of the May 22, 2017 Manchester Arena bombing, the latter his brother who was found guilty of aiding Saleem (Parveen & Walker, 2020), and the community ties surveyed by the US army in Thailand villages in 1965 (van Meter, 2002). The first two items thus inform creation of an edge between two individuals whereas the others inform edge creation as well as the edge weights between two individuals.

Note that it is important to differentiate two types of data associated with communications: the content of such communications and "secondary data", i.e. metadata such as the identities of parties and the timing, location and duration of communications. Often secondary data is available whilst content data is unavailable due to either encryption or limits prescribed by certain interception warrants. Moreover due to technology companies' planned future adoption of encryption for a wider range of communication technologies, the availability to investigators of content data is likely to decrease (Watney, 2020). However, even secondary data without content data has proven to be extremely useful: "so-called *secondary data* can enable the tracing of contacts, associations, habits and preferences" (Anderson, 2016). Our model assumes at a minimum some availability of secondary data. Whilst content data is not required it can be utilised when available in informing the creation of edges and edge weights in the dynamic network model and modification of task intensities in the RVE model.

6.5.2 Notation

As described earlier, let Ω_t^* be the open population of POIs at time t and let $\Omega_t \subseteq \Omega_t^*$ be the subset of individuals that the authorities have decided to investigate and monitor at time t. Further, recall that typically routine monitoring data collected by the policing authorities are likely to be recorded at fixed time intervals. Here we are modelling the extent of information that can be shared between any two individuals within our network. For this purpose, we find it sufficient to treat the time t as discrete. For example, the discrete time steps can be hourly, daily or weekly depending on the granularity that best suits the observation process. While unexpected observations may arrive irregularly between the

discrete time steps, unlike in the RVE model, we find that the discretisation is unlikely to lead to significant loss of information due to the scope of the network model and the fact that in practice, its outputs are meant to be used in conjunction with the RVE models of the individuals forming the network, as described in Section 6.6.

For concreteness, note that the size of Ω_t may be in the hundreds and Ω_t^* in the thousands (Anderson, 2016). During each time period, new leads are discovered. From among these leads, new investigative cases are opened for those that pass a triage process meeting defined criteria. This typically includes the following aspects: (i) Risk: is there any evidence of risk in intelligible form, (ii) Credibility: is the information reliable; (iii) Actionable: can anything actually be done about it; (iv) and Proportionality: is investigation of the lead necessary and proportionate within legal and statutory obligations, resources and priorities (Anderson, 2016; Kirk-Wade & Allen, 2020). The triage process thus gives rise to a set of newly identified individuals at each time interval. Denote by Ω_t^+ the individuals who join the set Ω_t during a time interval. Over the same interval, a set Ω_t^- are lost from Ω_t for a variety of reasons as discussed in Section 6.3. For simplicity, assume that Ω_t^+ join the set Ω_t at the start of the time period t and existing individuals Ω_t^- are lost at the end of t. Thus we have

$$\Omega_t = \{\Omega_{t-1} \backslash \Omega_{t-1}^-\} \cup \Omega_t^+. \tag{6.6}$$

We create an undirected network $\mathcal{N}_t$ at each time t where

$$\mathcal{N}_t = (V(\mathcal{N}_t), E(\mathcal{N}_t)) \tag{6.7}$$

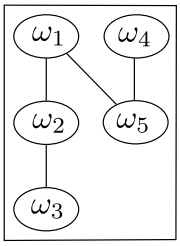
where $V(\mathcal{N}_t) = \Omega_t$ are the vertices and $E(\mathcal{N}_t)$ are the edges of the network. An edge exists $e_{ij} \in E(\mathcal{N}_t)$ between two individuals ω_i and ω_j if

1. They share an existing familial or social link;
2. They have committed crimes together in the past;
3. They have shared affiliations;
4. Since becoming POIs, they have been observed communicating with each other.

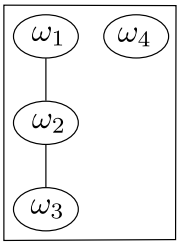
Once an edge is created in $\mathcal{N}_t$ between some ω_i, $\omega_j \in \Omega_t$, there exists an edge between them for all $\mathcal{N}_{t'}$ where $t' \geq t$. Let ϕ_{ijt} be the random variable measuring the *collaborative link*, i.e. the extent of information directly shared between ω_i and ω_j at time t. Thus ϕ_{ijt} gives the edge weight on the edge e_{ij} between ω_i and ω_j in $\mathcal{N}_t$. Denote by Φ_t a $|\Omega_t| \times |\Omega_t|$ symmetric matrix with its (i, j)th entry given by ϕ_{ijt}. By convention, we set $\phi_{ijt} = 0$ if $i = j$ or $e_{ij} \notin E(\mathcal{N}_t)$ for $i \neq j$. The pairwise communications data (Section 6.5.1) are used to infer ϕ_{ijt}. Below we introduce some notation for the pairwise communications data.

Example 6.9 (Criminal network)**.** *This example is borrowed from Bunnin et al. (2020). We consider here the investigative activities of a policing authority in a particular hypothetical town in the UK. The time steps here are assumed to be weekly. Four individuals from this town, namely* ω_1, ω_2, ω_3 *and* ω_4 *have been observed to have posted pro-terrorist material on social media and have been triaged into* Ω_t *the observed subpopulation at time t. Also at time t, a separate lead reveals the return of an individual* ω_5 *– who was known to previously have pro-terrorist ideas – from a country whose local terrorist groups are known to run large radicalisation campaigns targeting foreign individuals. Thus, the subpopulation of suspects under investigation at time t is given by* $\Omega_t = \{\omega_1, \omega_2, \omega_3, \omega_4, \omega_5\}$. *The preliminary investigation revealed that* ω_1 *and* ω_2 *attended the same secondary school and are the same age, and that* ω_2 *and* ω_3 *attend the same gym and are frequently seen together. Further,* ω_4 *and* ω_5 *were both known affiliates of, a now defunct, local criminal group, and* ω_1 *and* ω_5 *were arrested together for a minor offence in the past. Due to these pre-existing links, edges* $e_{1,2}$, $e_{2,3}$, $e_{4,5}$ *and* $e_{1,5}$ *can be created at time t, see Figure 6.3(a). In the duration of the week represented by time t,* ω_5 *has been arrested and extradited to another country, with which the UK shares an extradition agreement, on a serious accusation of kidnapping and murder. Thus,* ω_5 *is no longer a person of interest to the local policing authorities, and hence,* $\Omega_{t+1} = \{\omega_1, \omega_2, \omega_3, \omega_4\}$. *The network at time* $t + 1$ *is represented in Figure 6.3(b). The authorities continue their monitoring activities on these individuals through the weeks represented by time* $t + 1$ *and* $t + 2$ *with no changes to the structure of the network, see Figures 6.3(b) and 6.3(c). At time* $t + 2$ *it is discovered that mobile phones newly registered to the addresses of* ω_1 *and* ω_4 *are in communication. This creates a tie between* ω_1 *and* ω_4 *as shown in Figure 6.3(d).*

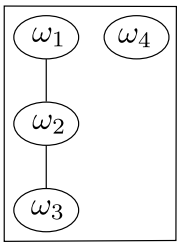
Recall that the policing authorities are likely to receive data from multiple channels. Suppose that there are m such *communication channels*. The data from each channel is recorded through a summary measure in our model. This summary measure can take a variety of forms depending on the type of data such as a sum of the observations (e.g. duration of phone calls or number of text messages exchanged) or a first-order difference (e.g. increase or decrease in money exchanged from one time to the next). Depending on the different communication channels, these summary measures can be on very different scales of measurement. For instance, the policing authorities might receive information from two communication channels: phone call records and bank transactions. Suppose that the phone calls are summarised in number of hours that the call lasted, and the bank transactions simply as the absolute value of the amount of money exchanged between the pair. In this case, x hours of a phone call is not equivalent to £x of money exchanged. To balance the effect on ϕ_{ijt} of data relating to different channels, the data obtained through the different channels needs to be on a comparable scale. This can be achieved through any of



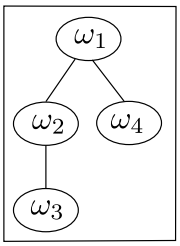
(a) At time t



(b) At time $t + 1$



(c) At time $t + 2$



(d) At time $t + 3$

Figure 6.3: Structure of network at times t, $t + 1$, $t + 2$ and $t + 3$.

the standard methods of scaling or normalisation (Jahan & Edwards, 2015).

Denote by s_{ijkt} the, possibly scaled/normalised, summary measure of the data observed between the pair $\omega_i, \omega_j \in \Omega_t$ from channel k at time t. We assume that the following independence relationship holds:

$$\perp\!\!\!\perp_{k\in\{1,\ldots,m\}} \; s_{ijkt} \tag{6.8}$$

which implies that the intelligence obtained from the various communication channels for a given pair $\{\omega_i, \omega_j\} \in \Omega_t \times \Omega_t$ at time t are mutually independent. Denote by U_t the *observations matrix* at time t with elements u_{ijt} such that $u_{ijt} = \{s_{ij1t}, \ldots, s_{ijmt}\}$. Notice that U_t is a symmetric $|\Omega_t| \times |\Omega_t|$ matrix with $u_{ijt} = u_{jit}$ due to the nature of the pairwise communications data. We use the convention that u_{ijt} is an m-dimensional zero vector whenever $i = j$, $e_{ij} \notin E(\mathcal{N}_t)$ for $i \neq j$, and whenever no information is observed between two individuals. To indicate the difference in the quality of data obtained from the different channels, we define a parameter $\xi_k \in (0, 1]$ which denotes the *efficiency* of the intelligence obtained from channel k, for $k = 1, \ldots, m$. This efficiency parameter indicates the loss of

information expected from a specific communication channel. A value closer to 1 represents minimal loss of information (e.g. information received about bank transitions), whereas a value closer to 0 indicates that the actual observations are likely to be much higher than what has been conveyed to the authorities (e.g. a patchy or poor source of secondary data).

6.5.3 The Steady Model

We now describe how the steady modelling technique (Smith, 1979, 1981) can be used to inform the edge weights modelled by the random variables ϕ_{ijt} for $\omega_i,\ \omega_j \in \Omega_t$ and $t \geq 0$. Typically, under non-Gaussian settings the derived one-step-ahead recurrences needed to update distributions of ϕ_{ijt} in our network are often not available in closed form. In such cases sampling or approximation would be required. Although sequential approximations such as those used in variational inference have some attraction, in our context, approximate methods would generally be prohibitive in terms of the computational demand and would also lead to some loss in transparency.

Here we adopt the approach of using a *steady model* from Gaussion dynamic linear models (M. West & Harrison, 1997). In a general sense, the steady model manipulates the posterior at time $t-1$ into a prior for time t such that the mean is kept constant or "steady" with a more diffuse variance. Here, we adapt the steady model into a non-Gaussian conjugate Gamma-Poisson setting (Smith, 1979, 1981). Interestingly, a different non-Gaussian variation of this model has also been applied to modelling online traffic flow count data in Chen et al. (2018).

To implement the steady model, we assume the following conditional independence relationships

$$\phi_{ijt} \perp\!\!\!\perp \mathcal{F}_t^- \,|\, \phi_{ij,t-1}, \tag{6.9}$$

$$u_{ijt} \perp\!\!\!\perp (\Phi_t, U_t, \mathcal{F}_t^-) \,|\, \phi_{ijt}. \tag{6.10}$$

where $\mathcal{F}_t^-$ denotes all past data and edge weight random variables up to but not including time t, i.e. $U_{t'}$ and $\Phi_{t'}$ for $t' < t$. Statement 6.9 is a standard first-order Markov assumption and Statement 6.10 signifies that the pairwise communications data between any pair of suspects $\{\omega_i, \omega_j\} \in \Omega_t \times \Omega_t$ at a given time t is only dependent on ϕ_{ijt}, the measure of information being exchanged between them at that time. With these standard assumptions in place, the relationship between the matrices U_t and Φ_t can then be represented by a 2-time-slice DBN (see Section 2.3.1) whose graph is shown in Figure 6.4. This enables us to estimate the collaborative link ϕ_{ijt} using observational data u_{ijt} for each pair ω_i and ω_j independently. To see how this links to the dynamic network model, recall that the estimates of $\phi_{\cdot,\cdot,t}$ inform the edge weights in the network model $\mathcal{N}_t$ at time t.

(a) (b)

Figure 6.4: Graphical representation of the 2-time-slice DBN: (a) on a univariate level; (b) on a multivariate level (labelled as graph $\mathcal{G}$).

Theorem 6.10. *The marginal likelihood of the 2-time-slice DBN represented by the graph $\mathcal{G}$ decomposes into the product of the one-step-ahead forecasts. Additionally all elements of U_t and Φ_t can be updated independently.*

Proof. At time $t > 0$, recall that $\mathcal{F}_t^-$ denotes all past data $U_{t'}$ and edge weight random variables $\Phi_{t'}$ for $t' < t$. Denote by $p(\phi_{ijt} \,|\, \mathcal{F}_t^-)$ the prior distribution for ϕ_{ijt} given the past information till time $t-1$. Since $\phi_{\cdot,\cdot,t}$ measures the extent of information being shared between a pair of suspects at time t, the random variable ϕ_{ijt} for each pair $\{\omega_i, \omega_j\} \in \Omega_t \times \Omega_t$ at time t can be estimated independently. Thus, the prior density of Φ_t can be written as

$$p(\Phi_t \,|\, \mathcal{F}_t^-) = \prod_{\{\omega_i,\, \omega_j\} \in \Omega_t \times \Omega_t} p(\phi_{ijt} \,|\, \mathcal{F}_t^-). \tag{6.11}$$

With the first-order Markovianity (Statement 6.9) and the output independence assumptions (Statement 6.10), the matrix U_t and the posterior of Φ_t decompose as follow

$$p(U_t \,|\, \mathcal{F}_t^-) = \prod_{\{\omega_i,\, \omega_j\} \in \Omega_t \times \Omega_t} p(u_{ijt} \,|\, \mathcal{F}_t^-), \tag{6.12}$$

$$p(\Phi_t \,|\, U_t, \mathcal{F}_t^-) = \prod_{\{\omega_i,\, \omega_j\} \in \Omega_t \times \Omega_t} p(\phi_{ijt} \,|\, u_{ijt}, \mathcal{F}_t^-). \tag{6.13}$$

For each pair $\{\omega_i,\ \omega_j\}$, on observing a vector u_{ijt} of data from the m communication channels, ϕ_{ijt} can be updated as follows due to Statement 6.8,

$$\begin{aligned} p(\phi_{ijt} \,|\, u_{ijt}, \mathcal{F}_t^-) &\propto p(\phi_{ijt} \,|\, \mathcal{F}_t^-) p(u_{ijt} \,|\, \phi_{ijt}, \mathcal{F}_t^-) \\ &= \prod_{k=1}^{m} p(\phi_{ijt} \,|\, \mathcal{F}_t^-) p(s_{ijkt} \,|\, \phi_{ijt}, \mathcal{F}_t^-). \end{aligned} \tag{6.14}$$

Thus the one-step-ahead forecasts can be written as

$$\begin{aligned} p(u_{ijt} \,|\, \mathcal{F}_t^-) &= \int_{\phi_{ijt}} p(u_{ijt} \,|\, \phi_{ijt}, \mathcal{F}_t^-) p(\phi_{ijt} \,|\, \mathcal{F}_t^-)\, d\phi_{ijt} \\ &= \prod_{k=1}^{m} \int_{\phi_{ijt}} p(s_{ijkt} \,|\, \phi_{ijt}, \mathcal{F}_t^-) p(\phi_{ijt} \,|\, \mathcal{F}_t^-)\, d\phi_{ijt} \end{aligned} \tag{6.15}$$

Now the marginal likelihood of the 2-time-slice DBN model described by the graph $\mathcal{G}$ can be decomposed into a product of the one-step-ahead forecasts as follows:

$$\begin{aligned} p(U_1, \ldots, U_t \,|\, \mathcal{F}_1^-) &= \prod_{s=1}^{t} p(U_s \,|\, \mathcal{F}_s^-) \\ &= \prod_{s=1}^{t} \prod_{\{\omega_i,\, \omega_j\} \in \Omega_s \times \Omega_s} \prod_{k=1}^{m} \int_{\phi_{ijs}} p(s_{ijks} \,|\, \phi_{ijs}, \mathcal{F}_s^-) p(\phi_{ijs} \,|\, \mathcal{F}_s^-)\, d\phi_{ijs} \end{aligned} \tag{6.16}$$

or equivalently as a sum of the log marginal likelihoods as

$$\log\Big(p(U_1, \ldots, U_t \,|\, \mathcal{F}_1^-)\Big) = \sum_{s=1}^{t} \sum_{\{\omega_i,\, \omega_j\} \in \Omega_s \times \Omega_s} \sum_{k=1}^{m} \log p(s_{ijks} \,|\, \mathcal{F}_s^-) \tag{6.17}$$

where $\mathcal{F}_1^-$ reflects the prior information used to set up the model at time t_0.

□

We now describe the Gamma-Poisson steady model below.

Initialisation: For each pair $\{\omega_i, \omega_j\} \in \Omega_t \times \Omega_t$, set the prior ϕ_{ijt_0} as follows

$$\phi_{ijt_0} \sim Gamma(\alpha_{ijt_0}, \beta_{ijt_0}) \tag{6.18}$$

where t_0 is the first time step in our time-series. The parameters α_{ijt_0} and β_{ijt_0} are determined by existing case knowledge. For example, if $e_{ij} \in E(\mathcal{N}_{t_0})$ exists only due to a social relation α_{ijt_0} and β_{ijt_0} may be set such that the mean and variance of ϕ_{ijt_0} are both relatively low. Whereas if ω_i and ω_j have a previous joint conviction then these parameters can be set such that the ϕ_{ijt_0} has a high mean and lower variance.

Posterior at time $t-1$: Let the posterior of $\phi_{ij,t-1}$ after observing $u_{ij,t-1}$ and $\mathcal{F}_{t-1}^-$ be given by

$$\phi_{ij,t-1} \,|\, u_{ij,t-1}, \mathcal{F}_{t-1}^- \sim Gamma(\alpha_{ij,t-1}, \beta_{ij,t-1}) \tag{6.19}$$

Prior at time t: Under the steady model, we use a discount factor $\delta_{ijt} \in (0, 1]$ to evolve the

posterior at time $t-1$ to the prior at time t as follows

$$\phi_{ijt} \,|\, \mathcal{F}_t^- \sim Gamma(\delta_{ijt}\alpha_{ij,t-1}, \delta_{ijt}\beta_{ij,t-1}). \tag{6.20}$$

The discount factor δ_{ijt} represents the decay of information from time $t-1$ to time t. The mean and variance of the prior are

$$\begin{aligned}
\mathbb{E}[\phi_{ijt} \,|\, \mathcal{F}_t^-] &= \frac{\delta_{ijt}\alpha_{ij,t-1}}{\delta_{ijt}\beta_{ij,t-1}} \\
&= \frac{\alpha_{ij,t-1}}{\beta_{ij,t-1}} \\
&= \mathbb{E}[\phi_{ij,t-1} \,|\, u_{ij,t-1}, \mathcal{F}_{t-1}^-]
\end{aligned} \tag{6.21}$$

$$\begin{aligned}
\text{var}[\phi_{ijt} \,|\, \mathcal{F}_t^-] &= \frac{\delta_{ijt}\alpha_{ij,t-1}}{(\delta_{ijt}\beta_{ij,t-1})^2} \\
&= \frac{\alpha_{ij,t-1}}{\delta_{ijt}(\beta_{ij,t-1})^2} \\
&\geq \frac{\alpha_{ij,t-1}}{\beta_{ij,t-1}^2} \\
&= \text{var}[\phi_{ij,t-1} \,|\, u_{ij,t-1}, \mathcal{F}_{t-1}^-]
\end{aligned} \tag{6.22}$$

Thus, the mean remains constant in the posterior to prior transformation and hence, this is known as a "steady" or stable evolution. The variance, however, either remains the same (when $\delta_{ijt} = 1$) or increases (when $0 < \delta_{ijt} < 1$). Thus, when the value of δ_{ijt} is close to one, then our prior for ϕ_{ijt} at time t is approximately identical to the posterior for $\phi_{ij,t-1}$. A lower value of δ_{ijt} indicates a reduced confidence in the posterior at the previous time step as the variance increases. This is also associated with a decay of information from the previous time step depending on how much the situation is likely to have evolved since then.

Data generation at time *t*: The observations from the different communication channels can be modelled independently (Statement 6.8). We assume that the observations from the communication channels are generated from a Poisson distribution;

$$s_{ijkt} \,|\, \phi_{ijt}, \mathcal{F}_t^- \sim Poi(\xi_k \phi_{ijt}), \quad k = 1, \ldots, m. \tag{6.23}$$

Note that this implies that the sample space of s_{ijkt} is the set of natural numbers including zero, $\mathbb{N}_0$ (see Appendix A).

Posterior at time *t*: The posterior when the observation vector u_{ijt} has at least one non-zero

element is given by

$$
\begin{aligned}
p(\phi_{ijt} \mid u_{ijt}, \mathcal{F}_t^-) &\propto \prod_{k=1}^{m} p(s_{ijkt} \mid \phi_{ijt}, \mathcal{F}_t^-)\, p(\phi_{ijt} \mid \mathcal{F}_t^-) \\
&= \prod_{k=1}^{m} \Big(\phi_{ijt}^{s_{ijkt}} \exp\big(-\xi_k \phi_{ijt}\big)\Big)\Big(\phi_{ijt}^{\delta_{ijt}\alpha_{ij,t-1}-1} \exp\big(-\delta_{ijt}\beta_{ij,t-1}\phi_{ijt}\big)\Big) \\
&= \phi_{ijt}^{\sum_k s_{ijkt}+\delta_{ijt}\alpha_{ij,t-1}-1} \exp\big(-(\textstyle\sum_k \xi_k + \delta_{ijt}\beta_{ij,t-1})\phi_{ijt}\big) \\
\phi_{ijt} \mid u_{ijt}, \mathcal{F}_t^- &\sim Gamma(\alpha_{ijt}, \beta_{ijt})
\end{aligned} \tag{6.24}
$$

where $\alpha_{ijt} = \delta_{ijt}\alpha_{ij,t-1} + \sum_k s_{ijkt}$ and $\beta_{ijt} = \delta_{ijt}\beta_{ij,t-1} + \sum_k \xi_k$. Observe here that for the same value of $\sum_k s_{ijkt}$, a lower overall efficiency of the observations given by $\sum_k \xi_k$ results in a higher mean and larger variance of ϕ_{ijt} compared to when the overall efficiency is higher. This is what we would expect as a lower efficiency indicates that the data we observe has some loss of information and hence, the actual extent of direct information exchanged (ϕ_{ijt}) is higher than the observed data indicates. The larger variance indicates the associated increase in uncertainty.

One-step-ahead forecast: The one-step-ahead forecast of the data from channel k can be obtained in closed form as

$$
\begin{aligned}
p(s_{ijk,t+1} \mid \mathcal{F}_{t+1}^-) &= \int_{\phi_{ij,t+1}} p(s_{ijk,t+1} \mid \phi_{ij,t+1}, \mathcal{F}_{t+1}^-)\, p(\phi_{ij,t+1} \mid \mathcal{F}_{t+1}^-)\, d\phi_{ij,t+1} \\
&= \int_{\phi_{ij,t+1}} \left\{ \frac{(\xi_k \phi_{ij,t+1})^{s_{ijk,t+1}} \exp\big(-\xi_k \phi_{ij,t+1}\big)}{s_{ijk,t+1}!} \right\} \times \\
&\qquad \left\{ \frac{(\delta_{ij,t+1}\beta_{ijt})^{\delta_{ij,t+1}\alpha_{ijt}} \phi_{ij,t+1}^{\delta_{ij,t+1}\alpha_{ijt}-1} \exp\big(-\delta_{ij,t+1}\beta_{ijt}\phi_{ij,t+1}\big)}{\Gamma(\delta_{ij,t+1}\alpha_{ijt})} \right\} d\phi_{ij,t+1} \\
&= \frac{\xi_k^{s_{ijk,t+1}} (\delta_{ij,t+1}\beta_{ijt})^{\delta_{ij,t+1}\alpha_{ijt}}}{s_{ijk,t+1}!\Gamma(\delta_{ij,t+1}\alpha_{ijt})} \times \\
&\qquad \int_{\phi_{ij,t+1}} \left\{ \phi_{ij,t+1}^{s_{ijk,t+1}+\delta_{ij,t+1}\alpha_{ijt}-1} \exp\big(-(\xi_k + \delta_{ij,t+1}\beta_{ijt})\phi_{ij,t+1}\big) \right\} d\phi_{ij,t+1} \\
&= \frac{\Gamma(s_{ijk,t+1} + \delta_{ij,t+1}\alpha_{ijt})\, (\delta_{ij,t+1}\beta_{ijt})^{\delta_{ij,t+1}\alpha_{ijt}}\, \xi_k^{s_{ijk,t+1}}}{\Gamma(\delta_{ij,t+1}\alpha_{ijt})\, (\xi_k + \delta_{ij,t+1}\beta_{ijt})^{(\delta_{ij,t+1}\alpha_{ijt}+s_{ijk,t+1})}\, s_{ijk,t+1}!} \\
&= \binom{s_{ijk,t+1} + \delta_{ij,t+1}\alpha_{ijt} - 1}{s_{ijk,t+1}} \frac{(\delta_{ij,t+1}\beta_{ijt})^{\delta_{ij,t+1}\alpha_{ijt}}\, \xi_k^{s_{ijk,t+1}}}{(\xi_k + \delta_{ij,t+1}\beta_{ijt})^{(\delta_{ij,t+1}\alpha_{ijt}+s_{ijk,t+1})}}
\end{aligned} \tag{6.25}
$$

where $\binom{\cdot}{\cdot}$ denotes the binomial coefficient.

In settings such as policing, it is essential to differentiate between the following cases:

1. $\sum_k s_{ijkt} = 0$ because ω_i and ω_j were monitored but did not communicate in any way during time t;
2. $\sum_k s_{ijkt} = 0$ because ω_i and ω_j were not closely monitored during time t.

In the first case, the posterior update is carried out as described above. In this case, we would expect the mean and variance to decrease. The posterior update results in this as shown below.

$$\begin{aligned}
\mathbb{E}[\phi_{ijt} \mid u_{ijt}, \mathcal{F}_t^-] &= \frac{\delta_{ijt}\alpha_{ij,t-1} + \sum_k s_{ijkt}}{\delta_{ijt}\beta_{ij,t-1} + \sum_k \xi_k} \\
&= \frac{\delta_{ijt}\alpha_{ij,t-1} + 0}{\delta_{ijt}\beta_{ij,t-1} + \sum_k \xi_k} \\
&< \frac{\delta_{ijt}\alpha_{ij,t-1}}{\delta_{ijt}\beta_{ij,t-1}} \\
&= \mathbb{E}[\phi_{ijt} \mid \mathcal{F}_t^-]
\end{aligned} \tag{6.26}$$

$$\begin{aligned}
\mathrm{var}[\phi_{ijt} \mid u_{ijt}, \mathcal{F}_t^-] &= \frac{\delta_{ijt}\alpha_{ij,t-1} + \sum_k s_{ijkt}}{(\delta_{ijt}\beta_{ij,t-1} + \sum_k \xi_k)^2} \\
&= \frac{\delta_{ijt}\alpha_{ij,t-1} + 0}{(\delta_{ijt}\beta_{ij,t-1} + \sum_k \xi_k)^2} \\
&< \frac{\delta_{ijt}\alpha_{ij,t-1}}{(\delta_{ijt}\beta_{ij,t-1})^2} \\
&= \mathrm{var}[\phi_{ijt} \mid \mathcal{F}_t^-]
\end{aligned} \tag{6.27}$$

as $\sum_k \xi_k > 0$.

In the second case, the prior at time t is set as the posterior at time t, i.e. a posterior update using the data is not carried out. This ensures that when we haven't actually *observed* zero communications, the posterior mean at time t is the same as the posterior mean at time $t-1$ and the posterior variance increases from time $t-1$ to t. In fact, in this case, the prior mean at time $t+1$ remains the same as the posterior mean at time $t-1$. However, the prior variance at time $t+1$ is further diffused as the posterior distribution at time $t-1$ is discounted twice to obtain the prior distribution at time $t+1$ as shown below:

$$\begin{aligned}
\mathbb{E}[\phi_{ij,t+1} \mid \mathcal{F}_{t+1}^-] &= \frac{\delta_{ij,t+1}\delta_{ij,t}\alpha_{ij,t-1}}{\delta_{ij,t+1}\delta_{ijt}\beta_{ij,t-1}} \\
&= \frac{\alpha_{ij,t-1}}{\beta_{ij,t-1}} \\
&= \mathbb{E}[\phi_{ij,t-1} \mid u_{ij,t-1}, \mathcal{F}_{t-1}^-]
\end{aligned} \tag{6.28}$$

$$
\begin{aligned}
\text{var}[\phi_{ij,t+1} \mid \mathcal{F}^{-}_{t+1}] &= \frac{\delta_{ij,t+1}\delta_{ijt}\alpha_{ij,t-1}}{(\delta_{ij,t+1}\delta_{ijt}\beta_{ij,t-1})^2} \\
&= \frac{\alpha_{ij,t-1}}{\delta_{ij,t+1}\delta_{ijt}(\beta_{ij,t-1})^2} \\
&\geq \frac{\alpha_{ij,t-1}}{\beta^2_{ij,t-1}} \\
&= \text{var}[\phi_{ij,t-1} \mid u_{ij,t-1}, \mathcal{F}^{-}_{t-1}]. \qquad (6.29)
\end{aligned}
$$

Observe that if no new information is observed through u_{ijs}, $s \geq t$ then the variance of ϕ_{ijs}, $s \geq t$ will continually keep increasing. To prevent this and to reflect that we expect a baseline amount of information flow to continue between a pair of suspects ω_i and ω_j who share an edge between them – until we observe information indicating otherwise – we can set the discount factor as $\delta_{ijt} = d_{ij} + (1 - d_{ij}) \exp\Big(-\sum_k s_{ijk,t-1}\xi_k\Big)$ (Chen et al., 2018). Here d_{ij} is the baseline discount factor for pair $\{\omega_i, \omega_j\}$. This is particularly useful if we expect to have large consecutive gaps of time when we do not expect to observe good quality data on the pairs. When we observe very low levels of quality information in the previous time, the discount factor is closer to 1 to limit the decay of the variance of ϕ_{ijt}. When good quality information is observed, the discount factor will be closer to d_{ij}. Notice that this setting allows us to set pair-specific discount factors if needed. However, in practise using a common discount factor across different pairs seems to work well. In the application presented in Section 6.7, we do not use a baseline discount factor and use a common discount factor across pairs for simplicity.

The distribution of ϕ_{ijt} for a pair $\{\omega_i, \omega_j\}$ can hence be periodically updated over the evolution of time t in closed form using the above recurrences across the network given the sequential incoming observational data. Also, the dynamic nature of the open population is easily incorporated in our model by introducing vertices, edges and priors for immigrants (new entrants) and removing them for emigrants (leavers) at the appropriate time. To summarise, Figure 6.5 gives an overview of the dynamic network model.

6.6 Integrating Decision Support System

In this section we illustrate how a collection of RVE models described in Section 6.3 and the dynamic weighted network model described in Section 6.5 can be combined together into an integrating decision support system (IDSS) for the criminal collaboration process. French et al. (2009) defines a decision support system as a "computer-based system which supports the decision making process, helping decision makers to form and explore the implications of their judgements and hence to make a decision based upon understanding". An IDSS was then defined by Leonelli and Smith (2015) as a unifying and integrating

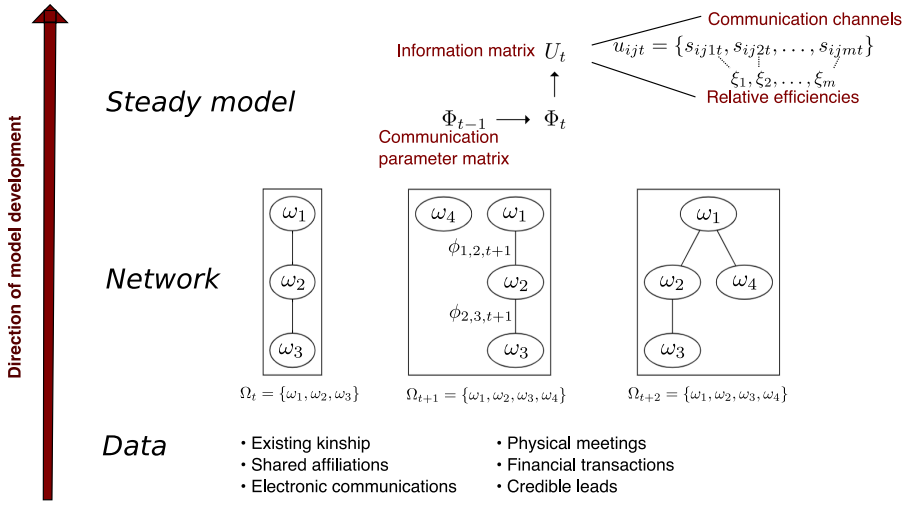


Figure 6.5: Overview of the dynamic network model.

framework that combines component decision support systems – each supporting decision making about a distinct aspect of a complex system – into a single entity.

We demonstrate how, with the appropriate conditional independence structure, we can decompose a complex dynamic multivariate system into independent dynamic univariate systems. With this, at each time $t > 0$, we can formally combine together the outputs of the network $\mathcal{N}_t$ and the individual RVE models for each individual ω in the population Ω_t while estimating the parameters of the network and RVE models independently.

6.6.1 The Decoupling Methodology

In this section we describe the decoupling methodology previously exploited to develop Multiregression Dynamic Models (MDMs) which were first introduced in Queen and Smith (1993) and later used for numerous applications, for example Freeman and Smith (2011b), Anacleto et al. (2013), Costa et al. (2015), Leonelli and Smith (2015), and Wilkerson and Smith (2021). We demonstrate how this decoupling methodology directly applies to the setting described in this chapter, thus enabling us to combine the dynamic network model and the individual RVEs to obtain the criminal collaboration model. We describe this decoupling methodology below after first introducing some standard time-series notation.

Denote by $\mathbf{Y}_t = \{Y_t(1), Y_t(2), \ldots, Y_t(n)\}$ a multivariate time-series composed of n components at time $t > 0$. Let $\mathbf{y}_t$ be the vector of observed values of all components of $\mathbf{Y}_t$ and $y_t(i)$ be the observed value of component $Y_t(i)$. Further, let $\mathbf{y}^t = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_t\}$,

$\mathbf{y}_i^t = \{y_1(i), y_2(i), \ldots, y_t(i)\}$ and $\mathbf{y}_A^t = \{\mathbf{y}_1(A), \mathbf{y}_2(A), \ldots, \mathbf{y}_t(A)\}$ where $A \subseteq \{1, 2, \ldots, n\}$. Denote the parameters associated with $\mathbf{Y}_t$ by $\boldsymbol{\theta}_t = \{\theta_t(1), \theta_t(2), \ldots, \theta_t(n)\}$ such that $\theta_t(i)$ is the parameter vector associated with component $Y_t(i)$. If the conditional independence structure of a dynamic model for $\mathbf{Y}_t$ can be represented by a DAG (see Section 2.2.1) with vertices representing the components of $\mathbf{Y}_t$, and the prior parameter vectors denoted by $\boldsymbol{\theta}_0$ are set to be mutually independent, then the following key results of Queen and Smith (1993) hold.

Proposition 6.11. *For a dynamic model over a time series $\boldsymbol{Y}_t = \{Y_t(1), Y_t(2), \ldots, Y_t(n)\}$ such that its conditional independence structure can be represented by a DAG whose vertices are the components of $\boldsymbol{Y}_t$, the following conditional independencies hold*

$$\perp\!\!\!\perp_{i \in [n]} \theta_t(i) \,|\, \boldsymbol{y}^t \tag{6.30}$$

$$\theta_t(i) \perp\!\!\!\perp \boldsymbol{y}^t_{[n] \setminus \{i \cup Pa(i)\}} \,|\, \boldsymbol{y}^t_i, \boldsymbol{y}^t_{Pa(i)} \tag{6.31}$$

where $[n] = \{1, 2, \ldots, n\}$ and such that $Pa(i)$ are the indices in $[n]$ associated with the parents of component $Y_t(i)$ in the DAG of the model.

The conditional independence in Statement 6.30 indicates that the parameter vectors for the different components remain independent for all time given the present and past observations, and the conditional independence in Statement 6.31 states that given the present and past observations for component $Y_t(i)$ and its parent components in the DAG of the model, the parameter vector for component $Y_t(i)$ is independent of the rest of the observed data (Queen & Smith, 1993; Leonelli, 2015). These conditional independencies ensure that the parameters associated with each component of the dynamic multivariate model can be updated independently at each time t and remain independent thereafter at each future time $t' > t$. With the specified DAG representation of the components of $\mathbf{Y}_t$, an MDM decomposes $\mathbf{Y}_t$ such that each of its components is a univariate dynamic linear model (DLM) (Harrison & Stevens, 1976; M. West & Harrison, 1997).

Theorem 2 of Queen and Smith (1993) proved the validity of Statements 6.30 and 6.31 for an MDM. However, we note here that this proof does not rely on each of the components being decomposed into univariate DLMs. In fact, the proof is an induction that simply relies on the prior parameter vectors $\boldsymbol{\theta}_0$ being mutually independent, and on an application of the d-separation theorem (see Section 2.3) on the DAG representation of the components of $\mathbf{Y}_t$. The key idea here is linking the components of $\mathbf{Y}_t$ through their observations directly rather than through their parameter vectors $\boldsymbol{\theta}_t$ within the DAG structure which enables us to preserve modularity of the time-series modelling of $\mathbf{Y}_t$ through all times $t > 0$.

The results given in Proposition 6.11 allow us to write the one-step-ahead forecast

distribution of $\mathbf{Y}_t$ as follows

$$p(\mathbf{y}_t \,|\, \mathbf{y}^{t-1}) = \prod_{i\in[n]} \int_{\theta_t(i)} f(y_t(i) \,|\, \mathbf{y}^t_{Pa(i)}, \mathbf{y}^{t-1}_i, \theta_t(i))\, p(\theta_t(i) \,|\, \mathbf{y}^{t-1}_{i\cup Pa(i)})\, d\theta_t(i), \tag{6.32}$$

which simplifies the joint forecast into a product of the individual forecasts for each component.

Example 6.12. *Consider a time series $\boldsymbol{Y}_t = \{Y_t(1), Y_t(2), Y_t(3), Y_t(4)\}$ with four components for $t > 0$. Suppose we model this time-series with an MDM and that the conditional independence structure associated with this model is given by the DAG in Figure 6.6. Then the time-series $\boldsymbol{Y}_t(i)$ associated with ith component is modelled as a univariate DLM. Further, the one-step-ahead forecast is given by*

$$\begin{aligned} p(\boldsymbol{y}_t \,|\, \boldsymbol{y}^{t-1}) = \int_{\theta_t(1)} & f(y_t(1) \,|\, \boldsymbol{y}^{t-1}_1, \theta_t(1))\, p(\theta_t(1) \,|\, \boldsymbol{y}^{t-1}_1)\, d\theta_t(1) \times \\ & \sum_{i\in\{2,3\}} \int_{\theta_t(i)} f(y_t(i) \,|\, \boldsymbol{y}^t_{i-1}\boldsymbol{y}^{t-1}_i, \theta_t(i))\, p(\theta_t(i) \,|\, \boldsymbol{y}^{t-1}_{\{i,i-1\}})\, d\theta_t(i) \times \\ & \int_{\theta_t(4)} f(y_t(4) \,|\, \boldsymbol{y}^t_{\{2,3\}}\boldsymbol{y}^{t-1}_4, \theta_t(4))\, p(\theta_t(4) \,|\, \boldsymbol{y}^{t-1}_{\{4,2,3\}})\, d\theta_t(4). \end{aligned} \tag{6.33}$$

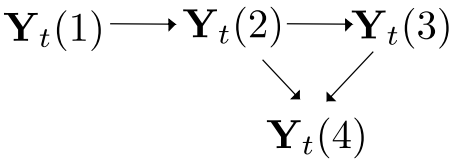


Figure 6.6: The DAG associated with the MDM in Example 6.12.

Leonelli (2015) and Leonelli and Smith (2015) describe in great detail, within the context of MDMs, how the modularity offered by the above decoupling methodology satisfy the necessary likelihood separation and independence conditions outlined therein for a distributed integrating decision support system (IDSS). Say that there is *structural consensus* (Leonelli, 2015; Leonelli & Smith, 2015) among the decision makers, users and other relevant stakeholders when they agree on the set of variables or components of $\mathbf{Y}_t$ together with a set of dependence statements about its components. Thus the structural consensus defines a class of models. Suppose now that a model satisfying the independence properties in Proposition 6.11 belongs to the structural consensus class of a particular process, and further suppose that a panel of experts G_j oversees the modelling of component set $\mathbf{Y}_t(B_j)$

where $B_j \subset [n]$, $j = 1, 2, \ldots, m$ and $B_j \cap B_k = \emptyset$ for $j \neq k$. Then the one-step-ahead forecasts for $\mathbf{Y}_t$ given in Equation 6.32 can be written as follows (Leonelli, 2015)

$$p(\mathbf{y}_t \,|\, \mathbf{y}^{t-1}) = \prod_{j \in [m]} \int_{\theta_t(B_j)} f(y_t(B_j) \,|\, \mathbf{y}^t_{Pa(B_j)}, \mathbf{y}^{t-1}_{B_j}, \theta_t(B_j))\, p(\theta_t(B_j) \,|\, \mathbf{y}^{t-1}_{B_j \cup Pa(B_j)})\, d\theta_t(B_j). \quad (6.34)$$

Hence, the decoupling methodology described here enables us to formally decouple the dynamic network models $\mathcal{N}_t$ and the individual RVE models of each $\omega \in \Omega_t$ for each time $t > 0$ and then recombine them within a modular integrating decision support system. In fact, this decoupling methodology, although well-demonstrated within MDMs, has not been exploited out of this setting until now. Recall that the properties of this methodology rely only on the initial independencies set through the prior parameters and on the DAG structure linking the components of the time-series. Hence, they can be easily transferred to suitable non-MDM settings.

6.6.2 IDSS of the Criminal Collaboration Model

We now describe how the above decoupling methodology can be applied to obtain our criminal collaboration model. Recall from Section 6.3 that in the RVE for a single suspect ω, $\mathbf{Y}_t$ refers to the data relating to the activities of ω at time $t > 0$. To generalise this notation to a population of suspects Ω_t, let $\mathbf{Y}_{it}$ denote the data relating to the activities of suspect $\omega_i \in \Omega_t$ at time $t > 0$. As defined in Section 6.5.2, let u_{ijt} be the m-dimensional vector containing summary measures of the information shared between individuals ω_i and ω_j through the m communication channels at time $t > 0$.

The dynamic network model $\mathcal{N}_t$ for population Ω_t can now be coupled with the $|\Omega_t|$ RVE models – one for each $\omega \in \Omega_t$ – through a DAG which contains edges from $\mathbf{Y}_{it}$ and $\mathbf{Y}_{jt}$ to u_{ijt} for each pair $\{\omega_i, \omega_j\} \in \Omega_t \times \Omega_t$, and no other edges. Recall that $u_{ijt} = u_{jit}$ by design and so we explicitly model only one of these within the DAG. For instance, consider $\Omega_t = \{\omega_i, \omega_j, \omega_k\}$. The DAG combining the individual RVEs for ω_i, ω_j and ω_k, and the dynamic network model $\mathcal{N}_t$ at time $t > 0$ is given in Figure 6.7.

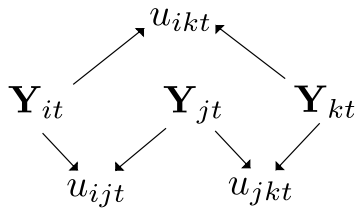


Figure 6.7: The DAG associated with the criminal collaboration model.

Further, since u_{ijt} contains all the observed information about the pairwise communications needed to estimate the edge weight modelled by random variable ϕ_{ijt} in the network model, and typically $u_{ijt} \subset Y_{it}$ and $u_{ijt} \subset Y_{jt}$, the estimation of ϕ_{ijt} can be performed independently of Y_{it} and Y_{jt} when u_{ijt} is given. Finally, by adapting Statement 6.10 as follows

$$u_{ijt} \perp\!\!\!\perp (\Phi_t, U_t, \{Y_{is}, Y_{js}\}_{s \leq t}, \mathcal{F}_t^-) \,|\, \phi_{ijt}, \tag{6.35}$$

the one-step-ahead forecasts of u_{ijt} as a product of the one-step-ahead forecasts of its s_{ijkt} components remain independent of Y_{it} and Y_{jt} when ϕ_{ijt} is given. This then allows estimation and forecasts for the RVE models and the networks models to be performed completely independently and then combined together to form the joint model.

Note, however, that the conditional independence in Statement 6.35 is a simplifying assumption. Within our application, it enables us to retain closed form recurrences and not make additional distributional assumptions. It is by no means a necessary assumption. We could instead incorporate the time-series corresponding to Y_{it} and Y_{jt} within the steady model for each pair $\{\omega_i, \omega_j\} \in \Omega_t \times \Omega_t$ in the dynamic network model or use a different model instead. Similarly, other improvements can be introduced if necessary. For instance, if we believed that the RVE model of a particular suspect ω_i was influenced by that of its neighbour ω_j (represented by an edge from Y_{jt} to Y_{it} in the DAG), then this can be easily incorporated too, for example by extracting the necessary signal from Y_{jt} with a filter (as described in the surface level of the RVE in Section 6.3) in the RVE model for Y_{it} or through addition of Y_{jt} as a regressor in a DLM for Y_{it}.

Through the combination of our bespoke dynamic network model and the existing RVE methodologies into an IDSS,

i) we have a formal stochastic model for modelling the threat posed by individuals – who form the vertices of our network model – informed by a Bayesian hierarchical model linking states, tasks and data;

ii) by defining our network model over an open population, our IDSS takes into account the periodic policing decisions of prioritising and de-prioritising cases;

iii) ties between individuals are modelled through the extent of information directly shared between them using observable data and prior information – these form the edges of our network model;

iv) we utilise a Bayesian paradigm to combine individual RVE processes and the multivariate dynamic network model;

v) we can define holistic threat scores to guide policing authorities in predicting and

disrupting real-time criminal activity, as described in Section 6.6.3.

6.6.3 Cell-Level Threat Scores

We now utilise the independent estimation of parameters across the criminal collaboration IDSS described above to arrive at a measure of imminence of threat posed by a known or suspected group of collaborating individuals within Ω_t. These groups are often called *cells*. There are various definitions of what constitutes a criminal or terrorist cell. Shapiro (2005) states that "a cell is best understood as an individual or group of individuals that take consequential actions". For our purposes we present a simple definition of a cell within our model as described below. A cell $C \subset \Omega_t$ is defined as a group of individuals who induce a connected subgraph in the network model $\mathcal{N}_t$ at time $t > 0$. The connectivity of the subgraph is to ensure that each individual in C is in contact with at least one other individual in C, failing which a joint effort would not be possible. The composition of C is typically determined by the investigators, based on their expert judgement, such that C is potentially an organisational unit for potential attack.

In addition to the dynamic network model which measures the flow of information between pairs of individuals within the population, to analyse the threat posed by a cell we also need to consider the threat posed by each member of the cell. The RVE model enables us to estimate the progression of an individual towards a particular attack. Hence, we combine the dynamic network model with the RVE models of the individuals forming a cell to measure the imminence of threat posed by the cell as a whole. This is equivalent to using the criminal collaboration IDSS for the stated purpose.

We first define the following notation

$$\begin{aligned}
\boldsymbol{\Theta}_t &= \{\boldsymbol{\theta}_{it} \,:\, \omega_i \in \Omega_t\} \in \{0,1\}^{R\times|\Omega_t|} \\
\boldsymbol{Z}_t &= \{\boldsymbol{Z}_{it} \,:\, \omega_i \in \Omega_t\} \in \mathbb{R}^{R\times|\Omega_t|} \\
\boldsymbol{W}_t &= \{W_{it} \,:\, \omega_i \in \Omega_t\} \in \{w_0, w_1, \ldots, w_n\}^{|\Omega_t|}
\end{aligned}$$

where $\boldsymbol{\theta}_{it}, \boldsymbol{Z}_{it}$ and W_{it} are defined as in Section 6.3 for the RVE of $\omega_i \in \Omega_t$.

The resources available to the authorities are typically limited and so it is critical that they are able to identify which cells pose the most imminent threat. To enable quick real-time support, we need threat scores that can be readily calculated from the available information and easily interpreted by the authorities. Based on this requirement, we present potential cell-level threat measures below which we later combine to arrive at informative threat scores.

(1) Collective progress: Similar to the individual RVE model described in Section

6.3, we can construct an RVE model for modelling the progress of the cell C, as a separate entity, towards a particular criminal attack. Let W_t^C indicate the latent random variable indicating the state occupied by the cell C at time $t > 0$, where $W_t^C \in \{w_0, w_1, \ldots, w_n\}$. As in the individual RVE, the data at the surface level $\mathbf{Y}_t^C = \cup_{\omega \in C} \mathbf{Y}_t^\omega$ is passed through a filter function to obtain $\mathbf{Z}_t^C$ which in turn informs the cell's engagement with the tasks $\boldsymbol{\theta}_t^C$ associated with the criminal attack. Denote by $\boldsymbol{\pi}_t^C = \{\pi_{t0}^C, \pi_{t1}^C, \ldots, \pi_{tn}^C\}$ the probability vector associated with the latent random variable W_t^C, such that π_{ti}^C indicates the probability that the cell is in state w_i at time $t > 0$. These can be obtained through the recurrences described in Section 6.3.1.

However, within a collaborative unit such as a cell, there will be some tasks that need only be done by a subset of the members of the cell; for example figuring out the logistics or developing certain skills. Thus, the filtered data $\mathbf{Z}_t^C$ obtained from the collective data on the cell $\mathbf{Y}_t^C$ must be set against these requirements to indicate whether the tasks are being *sufficiently completed*. Let $\boldsymbol{T}^C$ be the subset of the state space of W_t^C that indicates the set of states considered to be most dangerous by the authorities. A measure of *preparedness* of the cell can be obtained as

$$m_1 = \sum_{w_i^C \in \boldsymbol{T}^C} \pi_{ti}^C. \tag{6.36}$$

(2) Individual threat: Recall that the individual RVE represents the progress made by an individual on an attack they plan to enact by themselves. As discussed under the collective threat measure above, within a cell, not all tasks need to be performed by each and every member of the cell. Hence, if we consider the RVE of any individual member of the cell, for their progression towards the same criminal attack modelled for the entire cell above, they are likely to be further behind in their individual progress towards the attack than the cell as a whole. Thus, we would be underestimating the threat posed by each individual if we consider their individual RVEs for the same attack as the cell. Ideally we would be able to identify, for each member of C, the role that they play within the cell. Each role can then have an associated set of tasks which can be used within that individual's RVE model to monitor their progression towards the joint attack. However, this is not always possible as it requires detailed understanding of the cell's dynamics – intelligence which is extremely sensitive and difficult to gather (Duijn et al., 2014).

One option then is to evaluate the threat status of the individuals in C based on their progress on the tasks $\boldsymbol{\theta}_t^* \subset \boldsymbol{\theta}_t^C$ that *most of the members* of C are expected to have the skills to do. The states for the individual RVEs can be adapted in line with this to obtain the product of measures of *individual threat* for each member of C as

$$m_2 = \prod_{\omega \in C} \left\{ \sum_{w_i \in \boldsymbol{T}} \pi_{ti}^{\omega} \right\} \tag{6.37}$$

where $\boldsymbol{T}$ denotes the set of most dangerous threat states in the individual RVEs.

(3) Latent collaboration: In any cell, we may not expect each pair to be communicating with each other but for any successful collaborative project, a certain amount of connectivity is expected between each communicating pair and overall in the cell. Hence we set up two different measures of *cohesion*. For each communicating pair $\{\omega_i, \omega_j\}$ in C, we measure pairwise cohesion as

$$m_3^* = p(\phi_{ijt} > x_1) \tag{6.38}$$

where x_1 is the lower limit of how much we expect each pair to be communicating for the criminal attack to be enacted. A cell-level measure of pairwise cohesion can be obtained as

$$m_3 = \prod_{\{\omega_i, \omega_j\} \in \Omega_t \times \Omega_t} p(\phi_{ijt} > x_1). \tag{6.39}$$

Similarly, another cell-level cohesion measure can be obtained from the subnetwork *density* of C

$$m_4 = \frac{k}{\binom{n}{2}} \tag{6.40}$$

where $k = |E(C_t)|$ represents the number of ties shared by the members of cell C in the network model $\mathcal{N}_t$ at time $t > 0$, $n = |C_t|$ is the size of the cell C and thus $\binom{n}{2}$ is the number of possible ties in C.

(4) Size of the cell: While collaborative efforts benefit from sharing resources and skills, a very large cell can be unwieldy and increases the risk of exposure of the cell. For a given type of criminal attack, the authorities are likely to be able to estimate an ideal cell size p^* either from expert knowledge and intuition or from the literature and reported cases of a similar nature. A simple measure of *cell integrity* is obtained as

$$m_5 = \operatorname{sech}\left(\frac{p - p^*}{p^*} \right), \tag{6.41}$$

where $\operatorname{sech}(\cdot)$ is the hyperbolic secant function.

Cell threat scores: We can now combine the measures m_i for $i = \{1, 2, \ldots, 5\}$ to obtain

a series of cell-level summary measures called here as the *cell threat scores*. We caution against the use of any single summary measure as the sole determinant of the threat posed by a cell. Instead, these scores are meant to signal when the activities of the cell need to be monitored more closely.

For a given type of criminal attack, a cell is most threatening when $m_1 = m_2 = m_3 = m_4 = m_5 = 1$. We can obtain an ordered set of cell threat scores $\{\varphi_C(i)\}$, $i \in \{0, .., 4\}$ as

$$\varphi_C(i) = \prod_{j=1}^{5-i} m'_j \tag{6.42}$$
$$\{m'_j\}_{j=1,..,5} = \sigma(\{m_i\}_{i=1,..5})$$

where σ is a permutation of elements such that for $i = 1, \ldots 4$,

$$0 \leq m'_{i+1} \leq \ m'_i \leq 1$$

and hence for $i = 0, \ldots 3$,

$$0 \leq \varphi_C(i) \leq \varphi_C(i+1) \leq 1.$$

This ordered set is used to check whether a single or few measures' values are overly affecting the base $\varphi_C(0)$ score. Each of these scores has the property that a higher value of $\varphi_C(i)$ indicates a greater imminence and danger of the threat posed by the cell C. Thus we have combined several key factors to obtain transparent threat scores for a cell which can guide the authorities to prioritise and de-prioritise cases. These threat scores can be plotted against time to analyse how the threat posed by the cell develops dynamically. Note that the cell-level threat measures used here to define the cell threat scores can be easily adapted to incorporate other elements that may be considered to be essential by the policing authorities, see e.g. J. Xu et al. (2004) and Yang et al. (2006).

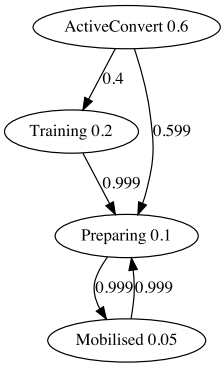
6.7 Review of a Simulated Example

In this section, we review a demonstration of the use of our criminal collaboration model as given in Bunnin et al. (2020). Due to the nature of policing, the required domain data is very sensitive and requires security clearance to acquire. Instead Example 6.9 was developed by Bunnin with simulated data to illustrate how the criminal collaboration model may work in practice. This data is simulated from some time t_1 which is equivalent to time $t + 1$ in Example 6.9, and is informed by meetings with relevant policing authorities and publicly available data on various real-world criminal cases.

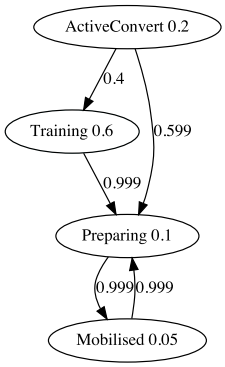
6.7.1 Individual RVEs

Recall that at time t_1 the individuals being monitored by the local policing authority are given by $\Omega_{t_1} = \{\omega_1, \omega_2, \omega_3, \omega_4\}$. Let the sample space of the latent random variable W_{it} in the RVE for suspect ω_i and time $t \geq t_1$ be given by the states {"Active convert", "Training", "Preparing", "Mobilised"}. Here, we shall refer to these states as *threat states*. Recall that the RDCEG of the RVE also includes a "Neutral" state that is not explicitly depicted.

Using the criminal profiles of the suspects, and based on their past and current activities, the prior probabilities of the state W_{i,t_1} occupied by these individuals at time t_1 are shown in the RDCEG graphs of the RVE models in Figure 6.8. Suspect ω_4 is believed to have received training by pro-terrorist groups and hence is placed in the "Training" state whereas the others have only stated their views and intentions but there is no indication otherwise of them training or preparing, hence they are placed in the "Active convert" state.



(a) RDCEG graph for ω_1, ω_2 and ω_3.

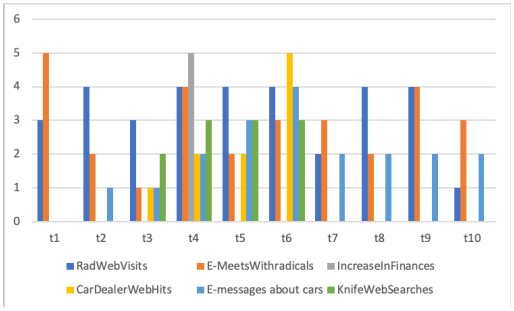


(b) RDCEG graph for ω_4.

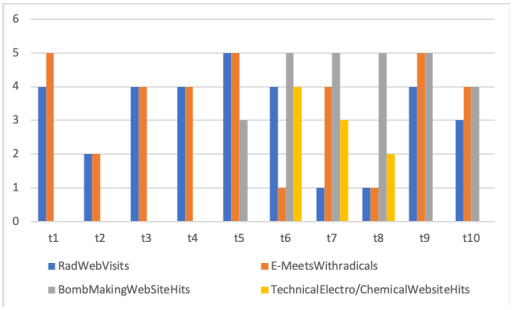
Figure 6.8: In both figures, the vertex labels include the prior state probability and edge labels denote the conditional transition probability at time t_1.

As these four individuals are in Ω_{t_1}, their activities and communications are monitored by the authorities. It is assumed for simplicity that over the ten weeks that follow, the composition of Ω_{t_1} remains unchanged, i.e. none of the existing suspects leave and no new suspects enter this subpopulation. Over the following weeks, it is observed that suspect ω_1's internet activities include repeated visits to websites of car dealers and car rentals, as well as knife retailers. Their bank account also shows a large influx of funds from an overseas bank account. The internet activity of suspect ω_2 includes visits to illegal bomb making websites, and repeated visits to and comments on extremist radical forums. Suspect ω_4's internet activity includes searches for online maps and blueprints of government buildings and densely populated commercial areas of the town. Suspect ω_4 is also observed to have

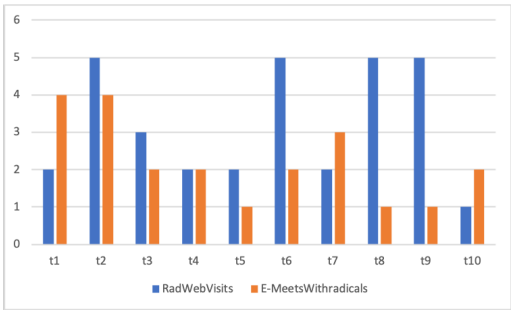
physically visited potential bomb testing sites. Figure 6.9 shows the activity data for each of the four suspects observed over a period of ten weeks. This data is used in the individual RVE models for the suspects as described in Section 6.3. Figure 6.10 shows the evolution of the posterior probabilities for the latent threat state variable W_{i,t_k} for suspect ω_i, $i = 1, 2, 3, 4$ and time $1 \leq k \leq 10$.



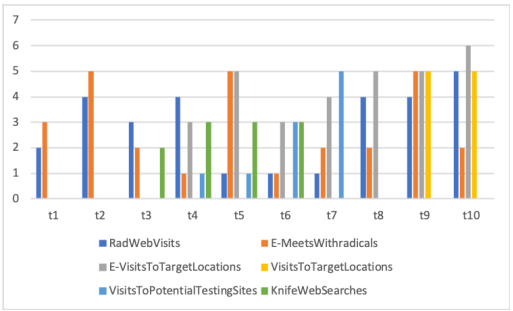
(a) Activity data for ω_1



(b) Activity data for ω_2



(c) Activity data for ω_3

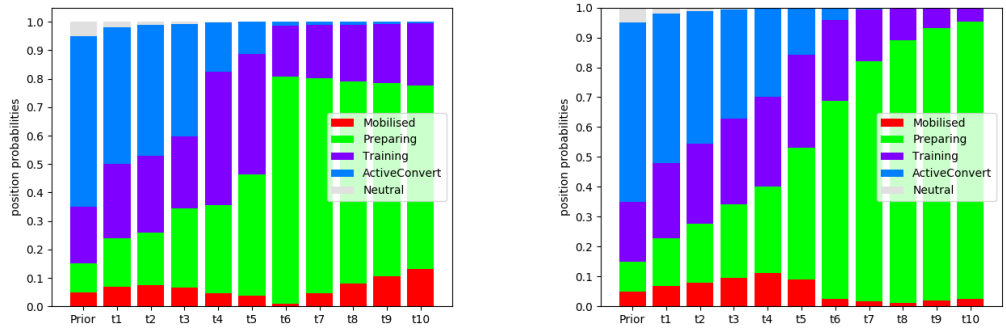


(d) Activity data for ω_4

Figure 6.9: Activity data for the four suspects over the observed time period of ten weeks.

6.7.2 Network Model

Not only are the activities of these suspects being observed, but their communications and interactions with each other are also being recorded over the ten week period. For simplicity, it is assumed here that the pairwise communications data are received from only one communication channel: mobile phone calls. The phone call data between a pair of suspects is summarised as the sum of the phone calls in hours between the pair observed over the week. Since we have only one communication channel here, we set its efficiency parameter as 1. Table 6.1 shows the summary data of the phone calls between each pair in our subpopulation over the ten week period. This data shows that during weeks t_2 and t_3, the pair $\{\omega_1, \omega_2\}$ talk over their mobile phones at a consistent rate. In week t_3, ω_1 and ω_4 are observed to converse with each other over a phone call. Further, in week t_5, we observe that the pairs $\{\omega_1, \omega_3\}$ and $\{\omega_2, \omega_4\}$ begin sharing phone conversations. These phone calls



(a) Posterior probabilities for ω_1

(b) Posterior probabilities for ω_2

(c) Posterior probabilities for ω_3

(d) Posterior probabilities for ω_4

Figure 6.10: Posterior threat state probabilities from the RVE models of the suspects over the ten weeks.

lead us to create edges $e_{1,3}$ and $e_{2,4}$ at time t_5 in the network to reflect these new ties. By week t_6, all four individuals share pairwise communications with each other. With this information, we can create edge $e_{3,4}$ in the network at time t_6 which results in the graph of the network becoming a complete graph. The total time of these phone calls are also observed to increase from weeks t_7 to t_{10}.

We can now use the Gamma-Poisson steady model described in Section 6.5.3 to model the evolution of the collaborative links among the four suspects. Recall that we denote this by ϕ_{i,j,t_k} for $i, j = 1, 2, 3, 4$, $i \neq j$ and $1 \leq k \leq 10$. The prior distributions for ϕ_{i,j,t_k} are set by specifying the α and β parameters of the prior Gamma distributions. For instance, based on the prior knowledge the policing authority has on the suspects, they believe that the extent of information shared between ω_1 and ω_2, and between ω_2 and ω_3 is relatively low with some uncertainty at time t_1. Based on this information, the α and β parameters are set as 0.7 and 1.41 respectively for $\phi_{1,2,t_1}$ and $\phi_{2,3,t_1}$. The setting of the α and β parameters for ϕ_{i,j,t_k} for all pairs over the ten week period are shown in Table 6.2, and

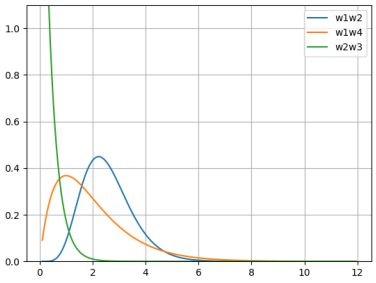
	$s_{1,2}$	$s_{1,3}$	$s_{1,4}$	$s_{2,3}$	$s_{2,4}$	$s_{3,4}$
t_1	0	0	0	0	0	0
t_2	3	0	0	1	0	0
t_3	5	0	2	0	0	0
t_4	5	0	5	0	0	0
t_5	5	2	5	0	1	0
t_6	5	6	6	5	6	1
t_7	7	6	7	6	7	7
t_8	6	6	8	4	8	8
t_9	7	7	9	7	9	9
t_{10}	7	8	11	8	10	10

Table 6.1: Simulated weekly sum of communication duration data. All the zeros in this table indicate that the pair did not communicate through mobile phone call in that week.

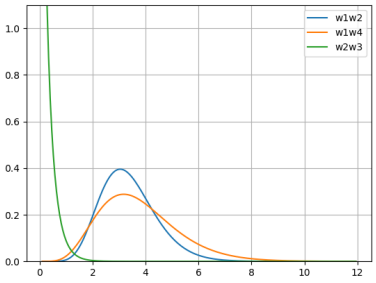
the evolution of ϕ_{i,j,t_k} is shown through the posterior densities in Figure 6.11. The discount factor $\delta_{i,j,t_k}$is set to 0.7 across all pairs and for the entire ten week duration.

	$\phi_{1,2}$		$\phi_{1,3}$		$\phi_{1,4}$		$\phi_{2,3}$		$\phi_{2,4}$		$\phi_{3,4}$	
	α	β	α	β	α	β	α	β	α	β	α	β
t_1 prior	0.70	1.41					0.70	1.41				
t_1 post	0.70	2.41					0.70	2.41				
t_2 prior	0.50	1.70					0.50	1.70				
t_2 post	3.50	2.70					1.50	2.70				
t_3 prior	2.46	1.90					1.05	1.90				
t_3 post	7.46	2.90			2	1	1.05	2.90				
t_4 prior	5.26	2.04			1.41	0.70	0.74	2.04				
t_4 post	10.26	3.04			6.41	1.70	0.74	3.04				
t_5 prior	7.23	2.15			4.52	1.20	0.52	2.15				
t_5 post	12.23	3.15	2	1	9.52	2.20	0.52	3.15	1	1		
t_6 prior	8.62	2.22	1.41	0.70	6.71	1.55	0.37	2.22	0.70	0.70		
t_6 post	13.62	3.22	7.41	1.70	12.71	2.55	5.37	3.22	6.70	1.70	1	1
t_7 prior	9.60	2.27	5.22	1.20	8.95	1.80	3.78	2.27	4.72	1.20	0.70	0.70
t_7 post	16.60	3.27	11.22	2.20	15.95	2.80	9.78	3.27	11.72	2.20	7.70	1.70
t_8 prior	11.70	2.30	7.91	1.55	11.24	1.97	6.89	2.30	8.26	1.55	5.43	1.20
t_8 post	17.70	3.30	13.91	2.55	19.24	2.97	10.89	3.30	16.26	2.55	13.43	2.20
t_9 prior	12.47	2.33	9.80	1.80	13.56	2.09	7.68	2.33	11.46	1.80	9.46	1.55
t_9 post	19.47	3.33	16.80	2.80	22.56	3.09	14.68	3.33	20.46	2.80	18.46	2.55
t_{10} prior	13.72	2.34	11.84	1.97	15.90	2.18	10.34	2.34	14.42	1.97	13.01	1.80
t_{10} post	20.72	3.34	19.84	2.97	26.90	3.18	18.34	3.34	24.42	2.97	23.01	2.80

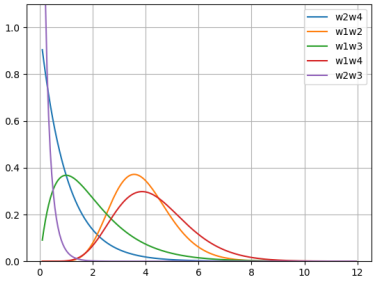
Table 6.2: Evolution of the prior and posterior parameters for ϕ_{i,j,t_k} during the ten weeks from t_1 to t_{10}.



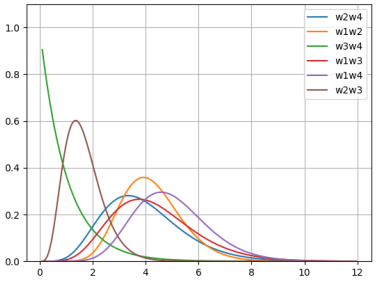
(a) Posterior density at t_3



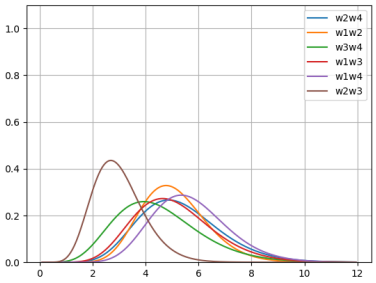
(b) Posterior density at t_4



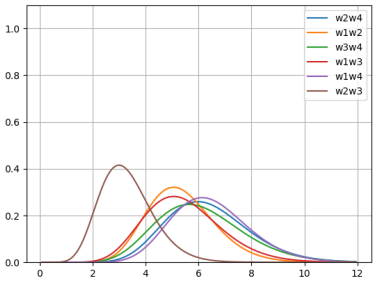
(c) Posterior density at t_5



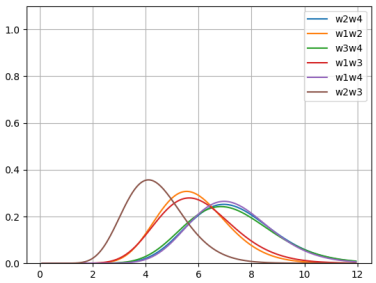
(d) Posterior density at t_6



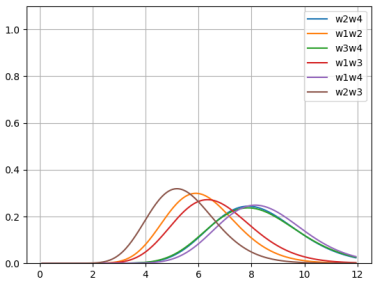
(e) Posterior density at t_7



(f) Posterior density at t_8



(g) Posterior density at t_9



(h) Posterior density at t_{10}

Figure 6.11: Evolution of ϕ_{i,j,t_k} from time t_3 to t_{10} represented through their posterior densities.

6.7.3 Cell-Level RVE Model and Threat Scores

Next we consider the threat posed by the suspects $\{\omega_1, \omega_2, \omega_3, \omega_4\}$ under the assumption that they might be working collaboratively as a criminal cell. With the criminal collaboration model – formed by combining together the individual RVEs and the network model described above – the cell-level threat measures m_i for $i = 2, 3, 4, 5$ are calculated as described in Section 6.6.3. Further, as discussed in Section 6.6.3 a cell-level RVE model is constructed for this cell where the task set and observation data are given by the union of the task sets and observation data for the individual RVEs of the cell's members. The prior threat state probabilities for the cell-level RDCEG is taken as the prior threat state probabilities of the suspect within the cell with the highest prior threat, i.e. ω_4. The task intensities for the cell-level RVE are based on the observational data of all the individuals in the cell. Figure 6.12 shows the evolution of the state probabilities in the RDCEG through time. As can be seen from this figure, the posterior probability of the cell being in the "Preparing" state increases from time t_5 as the communications within the cell and the overall activities of the cell increase. Thereafter around time t_9, the posterior probability of the cell being in the "Mobilised" state increases sharply.

The numeric cell-level threat measures m_1 to m_5 and the combined threat scores φ_C are shown in Table 6.3. If the policing authority choose to signal a warning when $\varphi_C(\cdot)$ reaches a certain threshold, say 0.15, then we can see that for $\varphi_C(0)$ this is not reached till time t_7 whereas for $\varphi_C(2)$ this is reached by time t_3. In practise, these measures and the chosen thresholds would need to be calibrated using domain experience and judgement.

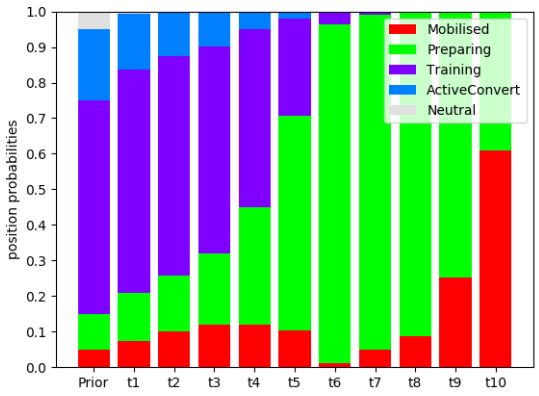


Figure 6.12: Posterior threat state probabilities from the cell-level RVE model over the ten weeks.

To summarise, this simple worked example demonstrates how observed activity data and communications data obtained on monitored suspects when combined with prior

	Prior	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
m1	0.15	0.21	0.26	0.32	0.45	0.71	0.96	0.99	1.00	1.00	1.00
m2	0.00	0.00	0.01	0.01	0.04	0.09	0.22	0.31	0.32	0.31	0.36
m3	0.14	0.05	0.14	0.04	0.03	0.00	0.30	1.00	1.00	1.00	1.00
m4	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.67
m5	0.83	0.83	0.83	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89
$\varphi_C(0)$	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.18	0.19	0.19	0.21
$\varphi_C(1)$	0.01	0.01	0.02	0.01	0.01	0.04	0.17	0.58	0.59	0.59	0.59
$\varphi_C(2)$	0.08	0.12	0.14	0.19	0.27	0.42	0.57	0.88	0.88	0.89	0.89
$\varphi_C(3)$	0.55	0.55	0.55	0.59	0.59	0.63	0.86	0.99	1.00	1.00	1.00
$\varphi_C(4)$	0.83	0.83	0.83	0.89	0.89	0.89	0.96	1.00	1.00	1.00	1.00

Table 6.3: Cell-level threat measures obtained through the criminal collaboration model and the cell-level RVE. The threat scores $\varphi_C(i)$ are defined as described in Section 6.6.3.

distributions calibrated to the investigator's knowledge can give real-time informative measures of the evolving threat posed by individuals acting in collaboration with others. The criminal collaboration model revises its probability estimates of the extent of information transfer within pairs of individuals as well as the latent threat state occupied by an individual and the cell in line with the incoming data. In the scenario investigated here, with synthetic data informed by real cases, our criminal collaboration model showed a marked increase in threat levels driven by the increase in specific activity data and phone call duration. This can be seen in the increased probability of the individuals forming the cell being in either the "Preparing" or "Mobilised" states by week t_{10}, and correspondingly, the cell as a whole appearing to move from state "Preparing" occupied since week t_5 to "Mobilised" by week t_{10}. The cell threat scores reflect a similar trend, e.g. $\varphi_C(2)$ increases from 0.27 during week t_4 to 0.57 during week t_6 and then reaching 0.88/0.89 for weeks t_7 to t_{10}.

6.8 Conclusion

In this chapter, we demonstrated how a special subclass of CT-DCEGs, called the RDCEG, can be useful in modelling processes on an open population where the missingness mechanism is likely to be MNAR. We also reviewed how the RDCEG forms an integral part of the three-level hierarchical RVE model presented in Bunnin and Smith (2019). We then proposed a two part criminal collaboration model with a novel dynamic weighted network model on one hand and the individual RVEs for each member of the extant monitored subpopulation on another. The novelty of the network model lies in modelling the weights along its edges – which measure the collaborative link between pairs of individuals – using

a steady model. With the steady model, we can not only take into account the temporal evolution of this pairwise measure of information exchanged even during the time periods where we may not observe any data by discounting our beliefs from the previous time periods, but also the recurrences are all in closed form. Further, we demonstrated how the decoupling methodology of MDMs can be seamlessly transferred to our setting to combine the two parts of our criminal collaboration model. Finally, we demonstrated how our criminal collaboration IDSS can inform the creation of simple yet powerfully informative threat scores that indicate the imminence of the threat posed by a potential criminal cell. The methods used in this chapter are novel to the policing domain application and provide a way of setting the literature on SNA methods applied to this domain within a more flexible and statistically defensible framework for further exploration.

There are several avenues of research that can follow from the work presented in this chapter. Below is a discussion on possible improvements to the model along with the statistical challenges they present, and domain challenges which our model currently does not address along with ideas of how we could begin to address them.

In our network, we create edges based on familial and kinship connections between pairs of individuals as well as past and present evidence of them sharing information with each other (see details in Section 6.5.2). Also recall that the edge weight ϕ_{ijt} along an edge in our model $\mathcal{N}_t$ is a measure of the extent of information being shared directly between the individuals ω_i and ω_j at time t connected by the edge. This definition of the edge weight then leads to the following conditional independence assumption:

$$\perp\!\!\!\perp_{\{\omega_i,\ \omega_j\}\in\Omega_t\times\Omega_t}\ \phi_{ijt}\,|\,\mathcal{F}_{t^-}. \tag{6.43}$$

Thus, pairwise communications data can be used to estimate the edge weight ϕ_{ijt}. For an alternative interpretation of the edge weights as measures of the extent of criminal collaboration between the individuals connected by the edge, the above conditional independence statement does not hold. This is because if we define ϕ_{ijt} to represent the extent of criminal collaboration between ω_i and ω_j, then this is clearly also affected by, for instance, the extent of collaboration they both share with a common neighbour ω_k. In this case, information from communications such as meetings, conference phone calls or other broadcasting methods involving three or more individuals will be informative of the extent of criminal collaboration between each of these pairs of individuals. This would additionally lead to the violation of the output independence assumption described in Statement 6.10. Under this independence structure, we would no longer be able to estimate ϕ_{ijt} for each pair ω_i and ω_j independently. One way to maintain some of the independence in the estimations while respecting this structure is given by the decouple/recouple strategy originally introduced

by Gruber and West (2016) and Zhao et al. (2016) for financial and economic multivariate time series applications. The decoupling/recoupling strategy involves defining a coherent multivariate dynamic model by coupling together sets of customised univariate dynamic models (M. West, 2020). In fact, MDMs and our IDSS could be classed as using the decouple/recouple strategy although our decoupling methodology described in Section 6.6.1 allows the recoupling to be done in closed form. However, the decoupling/recoupling strategy is useful even in cases where the recoupling cannot be done in closed form. Within our application, if ϕ_{ijt} measures the extent of criminal collaboration, we could decouple to independently model ϕ'_{ijt} which measures the extent of direct information exchanged and recouple these for all pairs $\{\omega_i, \omega_j\} \in \Omega_t \times \Omega_t$ to estimate ϕ_{ijt} in a way that respects the new independence structure (i.e. where Statements 6.10 and 6.43 do not hold). However, this is unlikely to be in closed form and MCMC methods will need to be applied. For applications using this kind of decoupling/recoupling strategy see, for example, Gruber and West (2017) and Chen et al. (2018). We then lose some amount of transparency in the model that we had because of the closed form recurrences. As we have noted earlier, within the domain of policing it is essential to maintain transparency and interpretability. So any gains achieved by incorporating the estimation of criminal collaboration between individuals in our model would have to weighed against the loss in transparency.

Sparrow (1991) points out that criminals may intentionally and successfully try to hide their activities and communications. Due to this, we may see very low levels of communication between certain individuals when, in fact, they are hiding or disguising their actual much higher levels of communication. Sparrow (1991) describes these as "weak ties". This implies that some connections where the level of activity is low might be of vital importance to the structural integrity of the network. Our criminal collaboration model is designed such that weak signals, within the RVEs of the individuals, will still be picked up with a well-designed filter function. Moreover, the model can be modified or overriden by the authorities at any point manually by changing the parameters, or creating or deleting ties as they see fit. The model works within a transparent Bayesian updating framework with initial inputs in the form of prior distributions and thus, all results can be traced back to the data and the quantitative assumptions.

However, certain links may be disguised well enough to be completely hidden from the policing authorities. Our network model, in its current form, does not include prediction of previously unknown links between individuals. Link prediction within SNA is a very active field and is clearly very pertinent to criminal network modelling. The literature contains several examples of predicting hidden links in criminal networks using existing SNA and emerging machine learning methodologies; see for example Budur et al. (2015), Berlusconi et al. (2016), Crandell and Korkmaz (2018), Lim, Abdullah, Jhanjhi, Khan, et al.

(2019), and Lim, Abdullah, Jhanjhi, and Supramaniam (2019). Link prediction methods, by design, make use of the network structure. Our model is not currently set up to use the network structure as within our current setting, it was sufficient to account for the temporal evolutions. Link prediction could be explored within our dynamic network model and would be particularly useful for the final task in the overarching Turing project of identifying new potential criminal groups as described in Section 6.1.

The final task of identifying previously unknown criminal groups or in short, community detection is not trivial within this domain. It is also rather different than in other social networks of a more benign nature; e.g. individuals in the network who have high measures of centrality are not necessarily the principal characters within a criminal group (Sparrow, 1991). It is essential to take into account the existing knowledge on the structure and dynamics of criminal networks along with the descriptions of the various roles of different individuals when considering community detection in these settings. J. J. Xu and Chen (2005), van Gennip et al. (2013), Ferrara et al. (2014), and D. Robinson and Scogings (2018) present examples of methodologies of community detection in criminal networks using SNA techniques. Bahulkar et al. (2018) is of notable mention as they perform edge augmentation through link prediction to identify hidden edges within the criminal network before using community detection methods. However, these methods do not take into account the criminality of the individuals in the network or the possible roles they play within their criminal groups. By developing appropriate stochastic set functions (as have been used in the literature, see for example Wang et al. (2013)) which can be used within our IDSS, perhaps after performing edge augmentation through link prediction in our dynamic network model, there is potential to develop a bespoke clustering algorithm to identify new criminal groups within the network.

Finally, the generic architecture of an IDSS using the decoupling methodology might be applicable to other domains where there is a requirement to integrate individual time-series with dynamic interactions among individuals, modelled by a network, who collaborate to realise a shared objective. Examples of this include social processes within politics, governments or communities where complex interacting individuals have shared objectives.

Chapter 7

Discussion

In Section 7.1 we summarise the main contributions of this thesis. In Section 7.2 we present a short discussion on two ongoing projects which have not been covered in the previous chapters. At the end of each research chapter we discussed, within the conclusion section, the contributions of that chapter along with a number of possible research directions that could follow thereon. In Section 7.3 we outline some additional research avenues.

7.1 Summary of the Contributions of this Thesis

This thesis presents the first documented systematic exploration of the non-stratified CEG class. This class of CEGs is what strongly differentiates it from the alternative BN family. Unlike BNs, non-stratified CEGs can not only explicitly represent context-specific conditional independencies within its graph topology but can also accommodate processes with asymmetric event spaces, typically arising due to structural zeros and structural missing values. Chapter 4 focused on defining and constructing a non-stratified CEG, and also presented an application of the class on a public health intervention. The examples used to illustrate the methods developed throughout this thesis all belong to the non-stratified class. In Chapter 5 we then explored and developed the CT-DCEG model class – a new general dynamic variant of CEGs that evolves in continuous time. In Chapter 6 we illustrated how a dynamic variant of the CEG family can be used alongside other models, each modelling a disparate component of an evolving complex system.

A more detailed description of the contributions is presented below:

- We demonstrated how CEGs – through their event tree construction – are an ideal framework for accommodating the two main issues that give rise to asymmetric event spaces structures, namely structural zeros and structural missing values.

- We presented a general backward iterative algorithm along with an optimal stopping criterion for transforming any staged tree, stratified or non-stratified, into a CEG.
- We formally proved that the mapping from a staged tree to a CEG is bijective, and hence, the CEG retains all the information represented by its staged tree.
- Through the modelling of a public health intervention to reduce falls-related injuries among the elderly, we demonstrated that the CEG is a superior model when compared to the BN for modelling processes with significant structural asymmetries.
- We presented the new dynamic model class of CT-DCEGs which evolves in continuous time and generalises the existing limited subclass of extended DCEGs. This class can embed crucial temporal information about the evolution of the process by explicitly modelling conditional holding time distributions for each event.
- We presented a model selection algorithm for a special subclass of the CT-DCEG class and illustrated how it can be applied to a dynamic extension of the falls intervention.
- With the bespoke semantics we developed for the CT-DCEG class, we presented a dynamic probability propagation scheme for this class. This scheme utilises two other developments presented in this thesis: 1) the extension CEG propagation algorithm such that it can incorporate temporal evidence, and 2) the approximate semi-Markov representation of any given CT-DCEG.
- We presented a special CT-DCEG model subclass called the RDCEG that conditions on individuals not dropping out of the population, and presented a detailed discussion on what it implies about the underlying missingness mechanism of the process.
- We developed a dynamic weighted network model with steady evolutions to model the extent of information being shared among a network of suspected criminals.
- We demonstrated how multiple disparate models can be easily combined into a complex multivariate dynamic IDSS, with the appropriate conditional independence assumption, by leveraging a decoupling methodology previously used for the MDM family of models.
- Finally, we presented a criminal collaboration IDSS that is obtained by combining the dynamic network model with a collection of RVE models – one for each member of the suspect pool – modelling an individual suspect's progression to a criminal activity. We illustrated how this IDSS can be used to create informative threat scores that indicate the progression of a criminal group to an attack.

7.2 Ongoing Work

7.2.1 Mixture Modelling Approach to Model Selection

As a CEG is completely defined by its underlying staged tree (see Theorem 4.9), model selection in CEGs is equivalent to identifying the collection of stages in its underlying event tree. Recall that thus far, the CEG literature has focused on identifying this collection of stages by 1) using a greedy AHC algorithm and 2) finding a globally optimal partition of the vertices with a dynamic programming approach. Even if we use the concept of hyperstages to reduce the model search space, we find that both of these approaches are not easily scalable. Additionally, while conjugacy of prior and posterior distributions of parameters is desirable for its analytical solutions and for the interpretability it lends to the hyperparameters, conjugate settings may either not be feasible or appropriate in certain cases. This is particularly of interest for CEGs with explicit modelling of holding times where the conditional holding time distributions may not have a conjugate prior (e.g. two-parameter Weibull distribution with unknown shape and scale parameters).

In light of both these issues, a CEG model selection approach that is scalable and does not rely on conjugate updating is useful. As part of a collaborative project with Dr Silvia Liverani, we propose a Bayesian mixture modelling approach to identifying the collection of stages within an event tree. Using the Stan programming language which is a current state-of-the-art technology we illustrate that the mixture model can be easily fit even when the conditional holding time distributions, and their priors are not conjugate.

We first briefly describe a finite mixture model before describing our proposed model selection method. For an excellent exposition of finite mixture models see Frühwirth-Schnatter (2006). Consider a population with K subgroups where each subgroup k is of relative proportion ℓ_k, for $k = 1, 2, \ldots, K$. Hence, $\sum_{k=1}^{K} \ell_k = 1$. Let $\boldsymbol{\ell} = \{\ell_1, \ell_2, \ldots, \ell_K\}$. Suppose that the interest lies in modelling a random feature Y such that Y is heterogeneous across the subgroups but homogeneous within each subgroup. Hence, each subgroup k can be associated with a parameter θ_k for the distribution modelling Y; i.e. the distribution of Y for subgroup k is given by $p(Y = y \mid \theta_k)$. Let $\boldsymbol{\theta} = \{\theta_1, \theta_2, \ldots, \theta_K\}$.

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

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

Let by $\mathbf{z} = \{z_1, z_2, \ldots, z_n\}$. Assuming random sampling from the population, the probability

that an individual belongs to subgroup i is given by the Categorical distribution $Cat(\boldsymbol{\ell})$.

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

$$\begin{aligned} p(\mathbf{y}) &= \prod_{i=1}^{n} p(y_i) \\ &= \prod_{i=1}^{n} \sum_{k=1}^{K} p(y_i, z_i^k) \\ &= \prod_{i=1}^{n} \sum_{k=1}^{K} p(z_i^k = 1|\boldsymbol{\ell})\ p(y_i \,|\, z_i^k = 1, \theta_k) \\ &= \prod_{i=1}^{n} \sum_{k=1}^{K} \ell_k p(y_i \,|\, \theta_k). \end{aligned} \tag{7.1}$$

For finite mixture models with more than one component (i.e. $K \geq 2$), the marginal likelihood $p(\mathbf{y} \,|\, \mathscr{M})$ for some model $\mathscr{M}$ is not available in closed form and must be numerically approximated (Frühwirth-Schnatter, 2006).

We next consider how the CEG model selection problem can be cast as a mixture modelling problem. Without loss of generality, consider a CEG with explicit modelling of conditional holding times. As in Section 5.7, notice that we can split the model selection process into two parts: 1) identifying the situation clusters, and 2) identifying the edge clusters.

Identifying the situation clusters

Consider an event tree $\mathcal{T}$ with n situations each with m outgoing edges and the same set of edge labels. For situation $s_i \in S(\mathcal{T})$, let its associated data vector be given by $\mathbf{y}_i = (y_{i1}, y_{i2}, \ldots, y_{im})$ where y_{ij} represents the number of individuals in the random sample that arrive at situation s_i and traverse its jth edge, for $1 \leq i \leq n$ and $1 \leq j \leq m$. Here $\mathbf{y} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n\}$ is the data vector and $\boldsymbol{\theta} = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_n\}$ is the parameter vector where $\boldsymbol{\theta}_i$ represents the conditional transition parameter vector for situation s_i. The model selection problem can be described as identifying the number and composition of the situation clusters in $\mathcal{T}$. This simplifies to fitting a standard finite mixture model as described by Equation 7.1 in Stan for a fixed number of components or situation clusters. However, we generally do not *a priori* know the number of situation clusters within a given event tree. A brute force solution involves fitting the finite mixture model in Stan with the number of components varying across all possible choices, i.e. from 1 to n for a total of n situa-

tions. A simplified pseudo-code of the proposed model selection algorithm is presented in Algorithm 5.

Algorithm 5: Mixture model selection algorithm for situation clusters

Input : Event tree $\mathcal{T}$, data $\mathbf{y}$, prior distribution for $\boldsymbol{\theta}_i$ for $1 \leq i \leq n$, prior distribution for $\boldsymbol{\ell}$.

Output: Optimal number of situation clusters, collection of situation clusters, log marginal likelihood score of the conditional transition parameters for the MAP CEG C found by the algorithm.

Set $allocation \leftarrow \emptyset$.
Set $parameters \leftarrow \emptyset$.
Set $score \leftarrow \emptyset$.
for K *from* 1 *to* n **do**
    Fit the model as described by Equation 7.1 in Stan with K components and do the following:
    Set $allocation_K$ as the composition of the K situation clusters given by the posterior allocation of each situation to one of the K components.
    $allocation \leftarrow allocation \cup allocation_K$
    Set $parameters_K$ as the parameters of the K situation clusters given by the posterior estimates of the parameters of each of the K components.
    $parameters \leftarrow parameters \cup parameters_K$
    Set $score_K$ as the log marginal likelihood of the model with K components.
    $score \leftarrow score \cup score_K$
Set $K(opt)$ = argmax $score$
return $allocation_{K(opt)}$, $parameters_{K(opt)}$, $score_{K(opt)}$

For an event tree which is stratified or has a hypercluster defined over its situations, the above algorithm can be run over each layer of the tree or each set within the hypercluster to identify the collection of situation clusters therein. While the above algorithm can easily handle several hundreds of situations for a fixed number of components, it will be significantly slowed down by fitting the mixture model for each possible number of components. The run time of the algorithm can be reduced by providing it with a smaller range of component numbers to explore.

In our preliminary experiments, we have found this algorithm to be very promising for situations with two emanating edges where the conditional transition parameter vector for each situation follows a Binomial distribution. However, for the case where a situation has three or more emanating edges, its conditional transition parameter vector follows a Multinomial distribution. Fitting a Multinomial finite mixture in Stan faces well-documented label switching problems among the components which results in identifiablility issues (Frühwirth-Schnatter, 2006; Mena & Walker, 2015).

Identifying the edge clusters

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

$$\begin{aligned} p(\mathbf{y}) &= \prod_{i=1}^{n} \sum_{k=1}^{K} \ell_k p(\mathbf{y}_i \,|\, \theta_k) \\ &= \prod_{i=1}^{n} \sum_{k=1}^{K} \ell_k \prod_{j=1}^{n_i} p(y_{ij} \,|\, \theta_k) \end{aligned} \tag{7.2}$$

for a fixed number of components or edge clusters K. The mixture model given by Equation 7.2 is non-standard because contrary to conventional mixture models, it does not imply that each data observation (i.e. each observation of a holding time for any edge) independently comes from one of the mixture components. Observe here that our model implies that all the observed holding times in $\mathbf{y}_i$ associated with edge e_i necessarily belong to the same component. In other words, all the observations in $\mathbf{y}_i$ are assumed to be drawn from the same distribution. Fortunately, fitting this model in Stan is straightforward. The pseudo-code for this algorithm is identical to the pseudo-code in Algorithm 5 with the except that the model to be fit in Stan is given by Equation 7.2. For an event tree with a hypercluster defined over its edges, this algorithm can be run over each set within the hypercluster. Preliminary experiments using two-parameter Weibull conditional holding time distributions where both the shape and scale parameters are unknown show promising results.

7.2.2 CEG Software

One possible hindrance to the wider application of CEG methodologies might be the lack of existing software. The popularity of the BN across a wide range of domains as a successful modelling instrument within an applied modeller's toolkit is influenced by the existence of several well-developed and regularly maintained software such as Netica, Weka,

BARD, GeNIe, and Hugin, as well as packages such as bnlearn, bayespy, GOBNILP, and BayeSuites. While there exist two R packages relevant to modelling with CEGs, namely ceg and stagedtrees, neither include the functionality needed to model non-stratified CEGs.

The CEG applications in this thesis were modelled using code I wrote in Python to support my methods. In its present form it supports modelling of non-stratified event trees, non-stratified CEGs and CT-DCEGs with model selection using the AHC algorithm. It also allows manual addition of edges with sampling zeros. The full development of this Python package is in progress, including added functionality to support CEG and CT-DCEG propagation algorithms, as well as model selection using the mixture methodology described in Section 7.2.1.

7.3 Future Work

We briefly outline three avenues of future research below:

- As the number of events or variables within a process grows, its event tree also grows. With this, we often face the issue of sparse edge counts where we observe very few individuals within our sample traversing certain edges of the event tree. Barclay (2014) provides a discussion of several possibilities of the criterion to determine that the counts for an edge are "sparse". Collazo and Smith (2016) has shown that the AHC, under Dirichlet local priors, tends to merge stages with high edge counts with those that have very low edge counts. This happens regardless of the true conditional distributions for these stages. Collazo and Smith (2016) also showed that product non-local priors generally do not face this issue. However, they are available in closed form only under certain conditions and are more computationally expensive to calculate. Another possible solution is to set an appropriate hyperstage structure that keeps situations that have very low counts within singleton sets to prevent any potentially spurious staging. However, this shifts the entire responsibility to the modeller to decide what the threshold should be for determining what counts as sparse. Further, this threshold would need to be calibrated to the application and the real-world cost associated with obtaining a spurious staging. Hence, it would be useful to investigate further ways in which greedy algorithms can be combined with appropriate prior settings (including product non-local priors) and score criteria to scale up model selection in the presence of sparse edge counts.

- Within the BN family, DBNs employ a single time granularity to model the evolution of a longitudinal process and alternatively, CTBNs model the evolution of a process in continuous time. For several heterogeneous complex systems, both these model

classes may fall short as DBNs may be insufficiently flexible to capture the dynamics of the system over a single time granularity whereas the temporal detail modelled by a CTBN may be too precise to match the available probabilistic knowledge. Liu et al. (2017) proposed a solution in the form of a new class of models called the hybrid time BN in which some variables evolve in discrete time over a fixed time granularity while others are modelled in continuous time. Thus, the hybrid time BN enables us to model complex systems with regularly and irregularly evolving random variables. Similar to this class of BNs, the work presented in Chapter 5 can be extended to develop hybrid time DCEGs. While most of the methodologies presented for CT-DCEGs should be directly transferable to this new model class, an appropriate alternative representation (analogous to SMPs for CT-DCEGs) for such hybrid time models would need to be explored.

- Finally, the last decade has seen rapid development of CEG technologies. CEGs have now been shown to have a flexible framework for modelling processes with asymmetric evolutions and asymmetric independence structures. There exist methodologies to support parameter learning, model selection, inference and reasoning in CEGs and its dynamic variants. While these methodologies have scope for improvement and for further development, it is now essential that more real-world applications of CEGs are explored. The literature thus far, including this thesis, present simulated case studies and some limited use of real-world data in domains such as public health, medicine, security and policing, reliability engineering, public services and education. The next step should be to explore more applications of CEGs in these and other domains. Directions for further methodological research are also likely to arise naturally through the process of modelling with CEGs in diverse domains.

Appendix A

Probability Distributions

The forms of the probability mass functions (pmf) or probability density functions (pdf) of the probability distributions that feature in this thesis are provided below (in alphabetical order) for reference.

Binomial distribution: The pmf of a random variable X that follows a Binomial distribution $Bin(n, p)$ with parameters $n \in \{0, 1, 2, \ldots\}$ denoting number of trials and $p \in [0, 1]$ denoting the probability of success of each trial is given as

$$p(X = x \,|\, n, p) = \binom{n}{x} p^x (1 - p)^{n-x}, \tag{A.1}$$

where $\binom{\cdot}{\cdot}$ represents the Binomial coefficient. The support of X is $\{0, 1, \ldots, n\}$.

Categorical distribution: The pmf of a random variable X that follows a Categorical distribution $Cat(p_1, p_2, \ldots, p_k)$ with parameters $p_i > 0$ denoting the probability of the random variable belonging to the ith category, for $1 \leq i \leq k$ is given by

$$p(X = i \,|\, p_1, p_2, \ldots, p_k) = p_i, \tag{A.2}$$

where $\sum_{i=1}^{k} p_i = 1$. The support of X is $\{1, 2, \ldots, k\}$.

Dirichlet distribution: The pdf of a set of random variables $\{X_1, X_2, \ldots, X_n\}$, for $n \geq 2$, that follows a Dirichlet distribution $Dir(\alpha_1, \alpha_2, \ldots, \alpha_n)$ with concentration parameters $\alpha_i > 0$ where $1 \leq i \leq n$ is given as

$$p(x_1, x_2, \ldots, x_n \,|\, \alpha_1, \alpha_2, \ldots, \alpha_n) = \frac{\Gamma(\sum_{i=1}^{n} \alpha_i)}{\prod_{i=1}^{n} \Gamma(\alpha_i)} \prod_{i=1}^{n} x_i^{\alpha_i - 1}, \tag{A.3}$$

where $\{x_i\}_{i=1}^n$ belongs to an $n-1$ dimensional simplex, i.e. $x_i \geq 0$ and $\sum_{i=1}^n x_i = 1$. The support of X_i is $(0, 1)$.

Exponential distribution: The pdf of a random variable X that follows an Exponential distribution $Exp(\lambda)$ with rate parameter $\lambda > 0$ is given as

$$p(X = x \,|\, \lambda) = \lambda \exp(-\lambda x). \tag{A.4}$$

The support of X is $[0, \infty)$.

Gamma distribution: The pdf of a random variable X that follows a Gamma distribution $Gamma(\alpha, \beta)$ with shape parameter $\alpha > 0$ and rate parameter $\beta > 0$ is given as

$$p(X = x \,|\, \alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} \exp(-\beta x). \tag{A.5}$$

The support of X is $(0, \infty)$.

Inverse-Gamma distribution: The pdf of a random variable X that follows an Inverse-Gamma distribution $IG(\alpha, \beta)$ with shape parameter $\alpha > 0$ and scale parameter $\beta > 0$ is given as

$$p(X = x \,|\, \alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{-\alpha-1} \exp\Big(-\frac{\beta}{x}\Big). \tag{A.6}$$

The support of X is $(0, \infty)$.

Multinomial distribution: The Multinomial distribution is a generalisation of the Binomial distribution. It models the probability of counts for each of the $k \geq 2$ possible categories for n independent trials. The pmf of a set of random variables $\{X_1, X_2, \ldots, X_k\}$, for $k \geq 2$, that follow a Multinomial distribution $Mult(p_1, p_2, \ldots, p_n)$ with parameter $p_i > 0$ denoting the probability of a trial resulting in category i, for $1 \leq i \leq k$, is given by

$$p(x_1, x_2, \ldots, x_n \,|\, p_1, p_2, \ldots, p_k) = \frac{n!}{x_1! x_2! \ldots x_k!} \prod_{i=1}^{k} p_i^{x_i}, \tag{A.7}$$

where $\sum_{i=1}^k x_i = n$ and $\sum_{i=1}^k p_i = 1$. Here random variable X_i indicates the number of times the category i is the outcome in all the n trials. The support of X_i is $\{1, 2, \ldots, n\}$.

Poisson distribution: The pmf of a random variable X that follows a Poisson dis-

tribution $Poi(\lambda)$ with rate parameter $\lambda > 0$ is given as

$$p(X = x \mid \lambda) = \frac{\lambda^x \exp(-\lambda)}{x!}. \tag{A.8}$$

The support of X is $\{0, 1, 2, \ldots\}$.

Weibull distribution: The pdf of a random variable X that follows a Weibull distribution $Wei(\kappa, \lambda)$ with shape parameter $\kappa > 0$ and scale parameter $\lambda > 0$ is given as

$$p(X = x \mid \lambda) = \frac{\kappa}{\lambda} x^{\kappa-1} \exp\left(-\frac{x^\kappa}{\lambda}\right). \tag{A.9}$$

The support of X is $[0, \infty)$.

Appendix B

Infinite Tree Probability Measure

Theorem B.1. *A probability measure can be defined on the atoms of the event space generated by an infinite event tree.*

Proof. Consider an infinite event tree $\mathcal{T} = (V(\mathcal{T}), E(\mathcal{T}))$. For two event trees $\mathcal{T}_1, \mathcal{T}_2$ rooted at the same vertex v write $\mathcal{T}_1 \preceq \mathcal{T}_2$ if $\mathcal{T}_1$ is a subtree of $\mathcal{T}_2$. Say $\mathcal{T}_1$ is a *minimal coarsening* of $\mathcal{T}_2$ if $\mathcal{T}_1$ can be constructed from $\mathcal{T}_2$ by deleting exactly one floret. An infinite event tree can thus be constructed from a finite event tree by sequentially adding florets.

Let event tree $\mathcal{T} \triangleq (\mathcal{T}_1, \mathcal{T}_2, \ldots)$, where $\mathcal{T}_j \preceq \mathcal{T}_{j+1}$, such that each $\mathcal{T}_j$ is finite and $\mathcal{T}_j$ is a minimal coarsening of $\mathcal{T}_{j+1}$, $j \in \mathbb{N}$. For each tree $\mathcal{T}_j$ we can define a probability space given by $(\Omega_j, \mathscr{F}_j, \mathbb{P}_j)$.

For any tree $\mathcal{T}'$ with probability space $(\Omega', \mathscr{F}', \mathbb{P}')$, each path $\lambda \in \mathcal{T}'_\Lambda$ represents a possible outcome trajectory of the process modelled by $\mathcal{T}'$. Thus Ω' corresponds to $\mathcal{T}'_\Lambda$ and $\mathscr{F}'$ is the σ-algebra defined over Ω'.

The sample space Ω of $\mathcal{T}$ can then be written as an infinite product space given by $\Omega \coloneqq \Omega_1 \times \Omega_2 \times \ldots$, which has a product σ-algebra $\mathscr{F}$. As this product is countable, the σ-algebra $\mathscr{F}$ is generated by cylinder sets given by

$$C = \{(\omega_1, \omega_2, \ldots) \in \Omega : \omega_1 \in A_1, \omega_2 \in A_2, \ldots, \omega_k \in A_k\},$$

for some $k \in \mathbb{N}$ and $A_i \subseteq \mathscr{F}_i$ for $1 \leq i \leq k$. Each cylinder set describes an outcome trajectory for an individual from the root up to k transitions. The subsequent evolution of the outcome trajectory can be arbitrary. Each outcome trajectory then has an associated probability (trivially each root-to-leaf path of this finite tree can be assigned an equal probability). The probability of the cylinder set C is then given by

$$\mathbb{P}_C = \mathbb{P}_1(A_1) \times \ldots \times \mathbb{P}_k(A_k).$$

Note that the collection of finite unions of these cylinder sets forms an algebra $\mathscr{F}_0$. Thus $\mathscr{F}$ is the σ-algebra generated by $\mathscr{F}_0$.

Given the probability spaces $(\Omega_j, \mathscr{F}_j, \mathbb{P}_j)$, $j \in \mathbb{N}$, for a finite subset I of $\mathbb{N}$, let the product measure on Ω_I be denoted by $\mathbb{P}_I$. All $(\Omega_j, \mathscr{F}_j)$, $j \in \mathbb{N}$ are Borel spaces and the sequence of probability measures $\mathbb{P}_I$, $I \subset \mathbb{N}$ is a consistent family of finite-dimensional distributions by the construction given above. Then by Kolmogorov's Extension theorem there exists a unique probability measure $\mathbb{P}$ on the infinite product space Ω that agrees with the measures $\mathbb{P}_I$ on Ω_I, $I \subset \mathbb{N}$. □

Bibliography

2011 Census: Aggregate Data. (2011). http://dx.doi.org/10.5257/census/aggregate-2011-1

Aalen, O. O. (1987). Dynamic modelling and causality. *Scandinavian Actuarial Journal*, *1987*(3-4), 177–190.

Airoldi, E. M., Blei, D. M., Fienberg, S. E., & Xing, E. P. (2008). Mixed membership stochastic blockmodels. *Journal of Machine Learning Research*, *9*, 1981–2014.

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, *19*(6), 716–723.

Alaa, A. M., Hu, S., & van der Schaar, M. (2017). Learning from clinical judgments: Semi-Markov-modulated marked Hawkes processes for risk prognosis. *Proceedings of the 34th International Conference on Machine Learning*, 60–69.

Anacleto, O., Queen, C., & Albers, C. J. (2013). Multivariate forecasting of road traffic flows in the presence of heteroscedasticity and measurement errors. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, *62*(2), 251–270.

Anderson, D. (2016). Investigatory Powers Bill: Bulk powers review. https://www.gov.uk/government/publications/investigatory-powers-bill-bulk-powers-review

Ankinakatte, S., & Edwards, D. (2015). Modelling discrete longitudinal data using acyclic probabilistic finite automata. *Computational Statistics & Data Analysis*, *88*, 40–52.

Arroyo-Figueroa, G., & Sucar, L. E. (1999). A temporal Bayesian network for diagnosis and prediction. *Proceedings of the Fifteenth International Conference on Uncertainty in Artificial Intelligence*, 13–20.

Bahulkar, A., Szymanski, B. K., Baycik, N. O., & Sharkey, T. C. (2018). Community detection with edge augmentation in criminal networks. *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, 1168–1175.

Bangsø, O., & Wuillemin, P.-H. (2000). *Object oriented Bayesian networks: A framework for top-down specification of large Bayesian networks with repetitive structures* (tech. rep. Technical Report CIT-87.2-00-obphw1). Department of Computer Science, Aalborg University,

Barbu, V. S., & Limnios, N. (2008). *Semi-Markov chains and hidden semi-Markov models toward applications: Their use in reliability and DNA analysis*. Springer.

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

Barclay, L. M., Hutton, J. L., & Smith, J. Q. (2013). Refining a Bayesian network using a chain event graph. *International Journal of Approximate Reasoning*, *54*(9), 1300–1309.

Barclay, L. M., Hutton, J. L., & Smith, J. Q. (2014). Chain event graphs for informed missingness. *Bayesian Analysis*, *9*(1), 53–76.

Barclay, L. M. (2014). *Modelling and reasoning with chain event graphs in health studies* (Doctoral dissertation). The University of Warwick.

Bauckhage, C. (2013). *Computing the Kullback-Leibler divergence between two Weibull distributions* [arXiv preprint arXiv:1310.3713].

Becker, G., Camarinopoulos, L., & Zioutas, G. (2000). A semi-Markovian model allowing for inhomogenities with respect to process time. *Reliability Engineering & System Safety*, *70*(1), 41–48.

Berlusconi, G., Calderoni, F., Parolini, N., Verani, M., & Piccardi, C. (2016). Link prediction in criminal networks: A tool for criminal intelligence analysis. *PlOS ONE*, *11*(4), 1–21.

Bhattacharjya, D., Shanmugam, K., Gao, T., Mattei, N., Varshney, K., & Subramanian, D. (2020). Event-driven continuous time Bayesian networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, *34*(04), 3259–3266.

Bhattacharjya, D., Subramanian, D., & Gao, T. (2020). State variable effects in graphical event models. *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, 4291–4297.

Billingham, L. J., & Abrams, K. R. (2002). Simultaneous analysis of quality of life and survival data. *Statistical Methods in Medical Research*, *11*(1), 25–48.

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

Box-Steffensmeier, J. M., & De Boef, S. (2005). Repeated events survival models: The conditional frailty model. *Statistics in Medicine*, *25*(20), 3518–3533.

Box-Steffensmeier, J. M., De Boef, S., & Joyce, K. A. (2007). Event dependence and heterogeneity in duration models: The conditional frailty model. *Political Analysis*, *15*(3), 237–256.

Boyen, X., & Koller, D. (1998). Tractable inference for complex stochastic processes. *Proceedings of the Fourteenth International Conference on Uncertainty in Artificial Intelligence*, 33–42.

Bozga, M., & Maler, O. (1999). On the representation of probabilities over structured domains. *International Conference on Computer Aided Verification*, 261–273.

Broccatelli, C., Everett, M., & Koskinen, J. (2016). Temporal dynamics in covert networks. *Methodological Innovations*, *9*.

Budur, E., Lee, S., & Kong, V. S. (2015). *Structural analysis of criminal network and predicting hidden links using machine learning* [arXiv preprint arXiv:1507.05739].

Bunnin, F. O., Shenvi, A., & Smith, J. Q. (2020). *Network modelling of criminal collaborations with dynamic Bayesian steady evolutions* [arXiv preprint arXiv:2007.04410].

Bunnin, F. O., & Smith, J. Q. (2019). A Bayesian hierarchical model for criminal investigations. *Bayesian Analysis*, *16*(1), 1–30.

Butler, R. W., & Huzurbazar, A. V. (2000). Bayesian prediction of waiting times in stochastic models. *Canadian Journal of Statistics*, *28*(2), 311–325.

Cameron, I. D., Murray, G. R., Gillespie, L. D., Robertson, M. C., Hill, K. D., Cumming, R. G., & Kerse, N. (2010). Interventions for preventing falls in older people in nursing care facilities and hospitals. *Cochrane Database of Systematic Reviews*, *12*.

Campedelli, G. M., Cruickshank, I., & Carley, K. M. (2019). A complex networks approach to find latent clusters of terrorist groups. *Applied Network Science*, *4*(1), 1–22.

Carli, F., Leonelli, M., Riccomagno, E., & Varando, G. (2020). *The R package stagedtrees for structural learning of stratified staged trees* [arXiv preprint arXiv:2004.06459].

Chan, H., & Darwiche, A. (2005). On the revision of probabilistic beliefs using uncertain evidence. *Artificial Intelligence*, *163*(1), 67–90.

Chen, X., Irie, K., Banks, D., Haslinger, R., Thomas, J., & West, M. (2018). Scalable Bayesian modeling, monitoring and analysis of dynamic network flow data. *Journal of the American Statistical Association*, *113*(522), 519–533.

Chickering, D. M. (1995). A transformational characterization of equivalent Bayesian network structures. *Proceedings of the Eleventh International Conference on Uncertainty in Artificial Intelligence*, 87–98.

Chickering, D. M., Heckerman, D., & Meek, C. (1997). A Bayesian approach to learning Bayesian networks with local structure. *Proceedings of the Thirteenth International Conference on Uncertainty in Artificial Intelligence* (pp. 80–89).

Clayton, D. G. (1991). A Monte Carlo method for Bayesian inference in frailty models. *Biometrics*, *47*(2), 467–485.

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

Collazo, R. A., & Smith, J. Q. (2016). A new family of non-local priors for chain event graph model selection. *Bayesian Analysis*, *11*(4), 1165–1201.

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

Collazo, R. A., & Smith, J. Q. (2018b). *Properties of an N time-slice dynamic chain event graph* [arXiv preprint arXiv:1810.09414].

Collazo, R. A., & Taranti, P. (2017). *ceg: Chain event graph* [R package version 0.1.0]. https://CRAN.R-project.org/package=ceg

Collazo, R. A. (2017). *The dynamic chain event graph* (Doctoral dissertation). The University of Warwick.

Cooper, G. F., & Herskovits, E. (1991). A Bayesian method for constructing Bayesian belief networks from databases. *Proceedings of the Seventh International Conference on Uncertainty in Artificial Intelligence*, 86–94.

Costa, L., Smith, J. Q., Nichols, T., Cussens, J., Duff, E. P., & Makin, T. R. (2015). Searching multiregression dynamic models of resting-state fMRI networks using integer programming. *Bayesian Analysis*, *10*(2), 441–478.

Cowell, R. G., Dawid, A. P., Lauritzen, S. L., & Spiegelhalter, D. J. (1999). *Probabilistic networks and expert systems*. Springer.

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

Crandell, I., & Korkmaz, G. (2018). Link prediction in the criminal network of Albuquerque. *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, 564–567.

Cunningham, D., Everton, S. F., & Murphy, P. J. (2015). Casting more light on dark networks: A stochastic actor-oriented longitudinal analysis of the Noordin top terrorist network. *Illuminating dark networks: The study of clandestine groups and organizations*, 171–185.

Cussens, J. (2008). Bayesian network learning by compiling to weighted MAX-SAT. *Proceedings of the Twenty-Fourth International Conference on Uncertainty in Artificial Intelligence*, 105–112.

Cussens, J. (2011). Bayesian network learning with cutting planes. *Proceedings of the Twenty-Seventh International Conference on Uncertainty in Artificial Intelligence*, 153–160.

Darwiche, A. (2003). A differential approach to inference in Bayesian networks. *Journal of the ACM*, *50*(3), 280–305.

Dawid, A. P. (1979). Conditional independence in statistical theory. *Journal of the Royal Statistical Society: Series B (Methodological)*, *41*(1), 1–15.

Dean, T., & Kanazawa, K. (1989). A model for reasoning about persistence and causation. *Computational Intelligence*, *5*(2), 142–150.

de Campos, L. M., & Castellano, J. G. (2007). Bayesian network learning algorithms using structural restrictions. *International Journal of Approximate Reasoning*, *45*(2), 233–254.

Departments of the Army and the Air Force. (1948). *Fundamentals of traffic analysis (Radio-Telegraph)* (tech. rep. US Manual number TM 32-250 - AFM 100-80). US Department of Defense.

Didelez, V. (2008). Graphical models for marked point processes based on local independence. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *70*(1), 245–264.

Dua, D., & Graff, C. (2019). UCI Machine Learning Repository. http://archive.ics.uci.edu/ml

Duijn, P. A., Kashirin, V., & Sloot, P. M. (2014). The relative ineffectiveness of criminal network disruption. *Scientific Reports*, *4*.

Edwards, D., & Ankinakatte, S. (2015). Context-specific graphical models for discrete longitudinal data. *Statistical Modelling*, *15*(4), 301–325.

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

Engelmann, N., Linzner, D., & Koeppl, H. (2020). Continuous time Bayesian networks with clocks. *Proceedings of the 37th International Conference on Machine Learning*, 2912–2921.

Epifani, I., Ladelli, L., & Pievatolo, A. (2014). Bayesian estimation for a parametric Markov renewal model applied to seismic data. *Electronic Journal of Statistics*, *8*(2), 2264–2295.

Europol, T. (2018). *EU terrorism situation and trend report* (tech. rep.). European Union Agency for Law Enforcement Cooperation.

Feller, W. (1971). *An introduction to probability theory and its applications*. John Wiley & Sons.

Fergusson, D. M., Horwood, L. J., & Shannon, F. T. (1986). Social and family factors in childhood hospital admission. *Journal of Epidemiology & Community Health*, *40*(1), 50–58.

Ferrara, E., De Meo, P., Catanese, S., & Fiumara, G. (2014). Detecting criminal organizations in mobile phone networks. *Expert Systems with Applications*, *41*(13), 5733–5750.

Fortunato, S., & Hric, D. (2016). Community detection in networks: A user guide. *Physics Reports*, *659*, 1–44.

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

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

French, S., Maule, J., & Papamichail, N. (2009). *Decision behaviour, analysis and support*. Cambridge University Press.

Friedman, N., & Goldszmidt, M. (1996). Learning Bayesian networks with local structure. *Proceedings of the Twelfth International Conference on Uncertainty in Artificial Intelligence* (pp. 252–262).

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

Galán, S. F., & Díez, F. J. (2002). Networks of probabilistic events in discrete time. *International Journal of Approximate Reasoning*, *30*(3), 181–202.

Geiger, D., & Heckerman, D. (1996). Knowledge representation and inference in similarity networks and Bayesian multinets. *Artificial Intelligence*, *82*(1–2), 45–74.

Geiger, D., Verma, T., & Pearl, J. (1990). D-separation: From theorems to algorithms. *Machine Intelligence and Pattern Recognition* (pp. 139–148). Elsevier.

Geneletti, S., & Dawid, A. P. (2011). Defining and identifying the effect of treatment on the treated. In P. M. Illari, F. Russo, & J. Williamson (Eds.), *Causality in the Sciences* (pp. 728–749). Oxford University Press.

Goffman, W. (1965). An epidemic process in an open population. *Nature*, *205*, 831–832.

Goldenberg, A., Zheng, A. X., Fienberg, S. E., & Airoldi, E. M. (2009). A survey of statistical network models. *Foundation and Trends in Machine Learning*, *2*(2), 129–233.

Gopalratnam, K., Kautz, H., & Weld, D. S. (2005). Extending continuous time Bayesian networks. *Proceedings of the 20th National Conference on Artificial Intelligence*, *2*, 981–986.

Görgen, C. (2017). *An algebraic characterisation of staged trees: Their geometry and causal implications* (Doctoral dissertation). The University of Warwick.

Görgen, C., & Smith, J. Q. (2016). A differential approach to causality in staged trees. *Proceedings of the Eighth International Conference on Probabilistic Graphical Models*, 207–215.

Görgen, C., & Smith, J. Q. (2018). Equivalence classes of staged trees. *Bernoulli*, *24*(4A), 2676–2692.

Gottard, A. (2007). On the inclusion of bivariate marked point processes in graphical models. *Metrika*, *66*(3), 269–287.

Gruber, L., & West, M. (2016). GPU-accelerated Bayesian learning and forecasting in simultaneous graphical dynamic linear models. *Bayesian Analysis*, *11*(1), 125–149.

Gruber, L., & West, M. (2017). Bayesian forecasting and scalable multivariate volatility analysis using simultaneous graphical dynamic models. *Econometrics and Statistics*, *3*, 3–22.

Grzegorczyk, M., & Husmeier, D. (2009). Non-stationary continuous dynamic Bayesian networks. *Proceedings of the 22nd International Conference on Neural Information Processing Systems*, 682–690.

Gunawardana, A., Meek, C., & Xu, P. (2011). A model for temporal dependencies in event streams. *Proceedings of the 24th International Conference on Neural Information Processing Systems*, 1962–1970.

Harper, W. R., & Harris, D. H. (1975). The application of link analysis to police intelligence. *Human Factors*, *17*(2), 157–164.

Harrison, P. J., & Stevens, C. F. (1976). Bayesian forecasting. *Journal of the Royal Statistical Society: Series B (Methodological)*, *38*(3), 205–228.

Heckerman, D. (1990). Probabilistic similarity networks. *Networks*, *20*(5), 607–636.

Heckerman, D. (2008). A tutorial on learning with Bayesian networks. *Innovations in Bayesian Networks*, 33–82.

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

Hoff, P. D., Raftery, A. E., & Handcock, M. S. (2002). Latent space approaches to social network analysis. *Journal of the American Statistical Association*, *97*(460), 1090–1098.

Howard, R. A. (1988). Decision analysis: Practice and promise. *Management Science*, *34*(6), 679–803.

Howard, R. A., & Matheson, J. E. (1981). Influence diagrams. In R. A. Howard & J. E. Matheson (Eds.), *The Principles and Applications of Decision Analysis* (pp. 721–762). Strategic Decision Group, Menlo Park, CA.

Hughes, D. M., Bonnett, L. J., Marson, A. G., & García-Fiñana, M. (2019). Identifying patients who will not reachieve remission after breakthrough seizures. *Epilepsia*, *60*(4), 774–782.

Investigatory Powers Act (c.25). (2016). https://www.legislation.gov.uk/ukpga/2016/25/contents/enacted

Iqbal, F., Fung, B. C., & Debbabi, M. (2012). Mining criminal networks from chat log. *2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, *1*, 332–337.

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

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

Jaeger, M., Nielsen, J. D., & Silander, T. (2006). Learning probabilistic decision graphs. *International Journal of Approximate Reasoning*, *42*(1–2), 84–100.

Jahan, A., & Edwards, K. L. (2015). A state-of-the-art survey on the influence of normalization techniques in ranking: Improving the materials selection process in engineering design. *Materials & Design (1980-2015)*, *65*, 335–342.

Janssen, J., & Manca, R. (2006). *Applied semi-Markov processes*. Springer.

Kalbfleisch, J. D. (1978). Non-parametric Bayesian analysis of survival time data. *Journal of the Royal Statistical Society: Series B (Methodological)*, *40*(2), 214–221.

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

Kirk-Wade, E., & Allen, G. (2020). *Terrorism in Great Britain: The statistics* (tech. rep.) [Briefing Paper Number CBP7613]. House of Commons Library. https://commonslibrary.parliament.uk/research-briefings/cbp-7613/

Kjærulff, U. (1992). A computational scheme for reasoning in dynamic probabilistic networks. *Proceedings of the Eighth International Conference on Uncertainty in Artificial Intelligence*, 121–129.

Koller, D., & Friedman, N. (2009). *Probabilistic graphical models: Principles and techniques*. MIT Press.

Koller, D., & Pfeffer, A. (1997). Object-oriented Bayesian networks. *Proceedings of the Thirteenth International Conference on Uncertainty in Artificial Intelligence*, 302–313.

Korb, K. B., & Nicholson, A. E. (2008). The causal interpretation of Bayesian networks. In D. E. Holmes & L. C. Jain (Eds.), *Innovations in Bayesian Networks, Studies in Computational Intelligence* (pp. 83–116). Springer.

Korb, K. B., & Nicholson, A. E. (2010). *Bayesian artificial intelligence*. CRC Press.

Krebs, V. E. (2002). Mapping networks of terrorist cells. *Connections*, *24*(3), 43–52.

Lauritzen, S. L. (1996). *Graphical models* (Vol. 17). Clarendon Press.

Lauritzen, S. L., & Wermuth, N. (1989). Graphical models for associations between variables, some of which are qualitative and some quantitative. *The Annals of Statistics*, 31–57.

Lee, L.-F., Liu, X., Patacchini, E., & Zenou, Y. (2012). Criminal networks: Who is the key player?

Leonelli, M. (2015). *Bayesian decision support in complex modular systems: An algebraic and graphical approach* (Doctoral dissertation). The University of Warwick.

Leonelli, M., & Smith, J. Q. (2015). Bayesian decision support for complex systems with many distributed experts. *Annals of Operations Research*, *235*, 517–542.

Lim, M., Abdullah, A., Jhanjhi, N., Khan, M. K., & Supramaniam, M. (2019). Link prediction in time-evolving criminal network with deep reinforcement learning technique. *IEEE Access*, *7*, 184797–184807.

Lim, M., Abdullah, A., Jhanjhi, N., & Supramaniam, M. (2019). Hidden link prediction in criminal networks using the deep reinforcement learning technique. *Computers*, *8*(1).

Little, R. J. A. (1988). A test of missing completely at random for multivariate data with missing values. *Journal of the American Statistical Association*, *83*(404), 1198–1202.

Little, R. J. A. (1993). Pattern-mixture models for multivariate incomplete data. *Journal of the American Statistical Association*, *88*(421), 125–134.

Little, R. J. A., & Rubin, D. B. (2019). *Statistical analysis with missing data* (3rd ed.). Wiley.

Little, R. J. A., Rubin, D. B., & Zangeneh, S. Z. (2017). Conditions for ignoring the missing-data mechanism in likelihood inferences for parameter subsets. *Journal of the American Statistical Association*, *112*(517), 314–320.

Liu, M., Hommersom, A., van der Heijden, M., & Lucas, P. J. (2017). Hybrid time Bayesian networks. *International Journal of Approximate Reasoning*, *80*, 460–474.

Liu, M., Stella, F., Hommersom, A., & Lucas, P. J. (2018). Making continuous time Bayesian networks more flexible. *Proceedings of the Ninth International Conference on Probabilistic Graphical Models*, 237–248.

Manrique-Vallier, D., & Reiter, J. P. (2014). Bayesian estimation of discrete multivariate latent structure models with structural zeros. *Journal of Computational and Graphical Statistics*, *23*(4), 1061–1079.

Meek, C. (2014). Toward learning graphical and causal process models. *Proceedings of the UAI 2014 Conference on Causal Inference: Learning and Prediction*, *1274*, 43–48.

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

Mohri, M., & Roark, B. (2005). *Structural zeros versus sampling zeros* (tech. rep.). Oregon Health & Science University, Portland, OR, USA.

Morse, J. M., & Field, P. A. (1996). *Nursing research: The application of qualitative approaches*. Nelson Thornes.

Morselli, C. (2009). *Inside criminal networks*. Springer.

Moura, M. d. C., & Droguett, E. L. (2008). A continuous-time semi-Markov Bayesian belief network model for availability measure estimation of fault tolerant systems. *Pesquisa Operacional*, *28*(2), 355–375.

Muliere, P., & Walker, S. (1997). A Bayesian non-parametric approach to survival analysis using Polya trees. *Scandinavian Journal of Statistics*, *24*(3), 331–340.

Nandy, S., Parsons, S., Cryer, C., Underwood, M., Rashbrook, E., Carter, Y., Eldridge, S., Close, J., Skelton, D., Taylor, S., & Feder, G. (2004). Development and preliminary examination of the predictive validity of the Falls Risk Assessment Tool (FRAT) for use in primary care. *Journal of Public Health*, *26*(2), 138–143.

Neapolitan, R. E. (2003). *Learning Bayesian networks*. Prentice Hall, Upper Saddle River, NJ.

NICE: Guidance and Guidelines. (2013). Falls in older people: Assessing risk and prevention. https://www.nice.org.uk/guidance/cg161

Nisbet, R. M., & Gurney, W. S. C. (1982). *Modelling fluctuating populations*. Wiley.

Nodelman, U., & Horvitz, E. (2003). *Continuous time Bayesian networks for inferring users' presence and activities with extensions for modeling and evaluation* (tech. rep. Techical Report MSR-TR-2003-97). Microsoft Research.

Nodelman, U., Shelton, C. R., & Koller, D. (2002). Continuous time Bayesian networks. *Proceedings of the Eighteenth International Conference on Uncertainty in Artificial Intelligence*, 378–387.

Nodelman, U., Shelton, C. R., & Koller, D. (2005). Expectation maximization and complex duration distributions for continuous time Bayesian networks. *Proceedings of the Twenty-First International Conference on Uncertainty in Artificial Intelligence*, 421–430.

Nurmi, I., & Lüthje, P. (2002). Incidence and costs of falls and fall injuries among elderly in institutional care. *Scandinavian Journal of Primary Health Care*, *20*(2), 118–122.

Office for National Statistics: 2011 Census Glossary. (2011). https://www.ons.gov.uk/census/2011census/2011censusdata/2011censususerguide/glossary

Pajouheshnia, R., Schuster, N. A., Groenwold, R. H., Rutten, F. H., Moons, K. G., & Peelen, L. M. (2020). Accounting for time-dependent treatment use when developing a prognostic model from observational data: A review of methods. *Statistica Neerlandica*, *74*(1), 38–51.

Parveen, N., & Walker, A. (2020). Brother of Manchester Arena bomber found guilty of murder. *The Guardian Newspaper*. https://www.theguardian.com/uk-news/2020/mar/17/brother-of-manchester-arena-bomber-hashem-abedi-guilty-murder

Pearl, J. (1986). Fusion, propagation, and structuring in belief networks. *Artificial Intelligence*, *29*(3), 241–288.

Pearl, J. (1994). A probabilistic calculus of actions. *Proceedings of the Tenth International Conference on Uncertainty in Artificial Intelligence*, 454–462.

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

Pearl, J., & Paz, A. (1986). Graphoids: Graph-based logic for reasoning about relevance relations or when would x tell you more about y if you already know z? *Proceedings of the 7th European Conference on Artificial Intelligence*, 357–363.

Phillips, L. D. (1984). A theory of requisite decision models. *Acta Psychologica*, *56*(1–3), 29–48.

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

Qin, Z., & Shelton, C. R. (2015). Auxiliary Gibbs sampling for inference in piecewise-constant conditional intensity models. *Proceedings of the Thirty-First International Conference on Uncertainty in Artificial Intelligence*, 722–731.

Queen, C. M., & Smith, J. Q. (1993). Multiregression dynamic models. *Journal of the Royal Statistical Society: Series B (Methodological)*, *55*(4), 849–870.

Ranciati, S., Vinciotti, V., & Wit, E. C. (2020). Identifying overlapping terrorist cells from the Noordin Top actor-event network. *Annals of Applied Statistics*, *14*(3), 1516–1534.

Robinson, D., & Scogings, C. (2018). The detection of criminal groups in real-world fused data: Using the graph-mining algorithm "GraphExtract". *Security Informatics*, *7*(2).

Robinson, J. W., & Hartemink, A. J. (2008). Non-stationary dynamic Bayesian networks. *Proceedings of the 21st International Conference on Neural Information Processing Systems*, 1369–1376.

Ron, D., Singer, Y., & Tishby, N. (1998). On the learnability and usage of acyclic probabilistic finite automata. *Journal of Computer and System Sciences*, *56*(2), 133–152.

Rubenstein, L. Z., Josephson, K. R., & Robbins, A. S. (1994). Falls in the nursing home. *Annals of Internal Medicine*, *121*(6), 442–451.

Rubin, D. B. (1976). Inference and missing data. *Biometrika*, *63*(3), 581–592.

Saria, S., Nodelman, U., & Koller, D. (2007). Reasoning at the right time granularity. *Proceedings of the Twenty-Third International Conference on Uncertainty in Artificial Intelligence*, 326–334.

Sarvari, H., Abozinadah, E., Mbaziira, A., & McCoy, D. (2014). Constructing and analyzing criminal networks. *2014 IEEE Security and Privacy Workshops*, 84–91.

Schafer, J. L., & Graham, J. W. (2002). Missing data: Our view of the state of the art. *Psychological Methods*, *7*(2), 147–177.

Schlaifer, R., & Raiffa, H. (1961). *Applied statistical decision theory*. MIT Press.

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, *6*(2), 461–464.

Schweder, T. (1970). Composable Markov processes. *Journal of Applied Probability*, *7*(2), 400–410.

Scutari, M. (2010). Learning Bayesian networks with the bnlearn R package. *Journal of Statistical Software*, *35*(3), 1–22.

Scutari, M. (2018). Dirichlet Bayesian network scores and the maximum relative entropy principle. *Behaviormetrika*, *45*(2), 337–362.

Sewall, W. (1921). Correlation and causation. *Journal of Agricultural Research*, 557–585.

Sewell, D. K., & Chen, Y. (2016). Latent space models for dynamic networks with weighted edges. *Social Networks*, *44*, 105–116.

Sewell, D. K., & Chen, Y. (2017). Latent space approaches to community detection in dynamic networks. *Bayesian Analysis*, *12*(2), 351–377.

Shachter, R. D. (1986). Evaluating influence diagrams. *Operations Research*, *34*(6), 819–953.

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

Shafer, G., & Shenoy, P. P. (1990). Probability propagation. *Annals of Mathematics and Artificial Intelligence*, *2*(1), 327–351.

Shapiro, J. N. (2005). *Organizing terror: Hierarchy and networks in covert organizations* (tech. rep.). Stanford University.

Shen, Y., Choi, A., & Darwiche, A. (2020). A new perspective on learning context-specific independence. *Proceedings of the Tenth International Conference on Probabilistic Graphical Models*.

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

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

Shenvi, A., & Smith, J. Q. (2020b). *Propagation for dynamic continuous time chain event graphs* [arXiv preprint arXiv:2006.15865].

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

Silander, T., Kontkanen, P., & Myllymäki, P. (2007). On sensitivity of the MAP Bayesian network structure to the equivalent sample size parameter. *Proceedings of the Twenty-Third International Conference on Uncertainty in Artificial Intelligence*, 360–367.

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

Silander, T., Roos, T., & Myllymäki, P. (2010). Learning locally minimax optimal Bayesian networks. *International Journal of Approximate Reasoning*, *51*(5), 544–557.

Smith, J. Q. (1979). A generalization of the Bayesian steady forecasting model. *Journal of the Royal Statistical Society: Series B (Methodological)*, *41*(3), 375–387.

Smith, J. Q. (1981). The multiparameter steady model. *Journal of the Royal Statistical Society: Series B (Methodological)*, *43*(2), 256–260.

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

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

Smith, J. Q., Barons, M. J., & Leonelli, M. (2015). *Coherent frameworks for statistical inference serving integrating decision support systems*.

Smith, J. Q., & Shenvi, A. (2018). *Assault crime dynamic chain event graphs* [Working Paper], University of Warwick. http://wrap.warwick.ac.uk/104824/

Sparrow, M. K. (1991). The application of network analysis to criminal intelligence: An assessment of the prospects. *Social Networks*, *13*(3), 251–274.

Spiegelhalter, D. J., & Lauritzen, S. L. (1990). Sequential updating of conditional probabilities on directed graphical structures. *Networks*, *20*(5), 579–605.

Spohn, W. (1980). Stochastic independence, causal independence, and shieldability. *Journal of Philosophical Logic*, *9*(1), 73–99.

Studený, M. (2005). *Probabilistic conditional independence structures*. Springer.

Sturlaugson, L., & Sheppard, J. W. (2016). Uncertain and negative evidence in continuous time Bayesian networks. *International Journal of Approximate Reasoning*, *70*, 99–122.

Su, J., & Zhang, H. (2005). Representing conditional independence using decision trees. *AAAI*, 874–879.

Thwaites, P. A. (2008). *Chain event graphs: Theory and application* (Doctoral dissertation). The University of Warwick.

Thwaites, P. A. (2013). Causal identifiability via chain event graphs. *Artificial Intelligence*, *195*, 291–315.

Thwaites, P. A., Smith, J. Q., & Cowell, R. G. (2008). Propagation using chain event graphs. *Proceedings of the Twenty-Fourth International Conference on Uncertainty in Artificial Intelligence*, 546–553.

Thwaites, P. A., Smith, J. Q., & Riccomagno, E. (2010). Causal analysis with chain event graphs. *Artificial Intelligence*, *174*(12–13), 889–909.

Toth, N., Gulyás, L., Legendi, R. O., Duijn, P., Sloot, P. M., & Kampis, G. (2013). The importance of centralities in dark network value chains. *The European Physical Journal Special Topics*, *222*(6), 1413–1439.

van Gennip, Y., Hunter, B., Ahn, R., Elliott, P., Luh, K., Halvorson, M., Reid, S., Valasik, M., Wo, J., Tita, G. E., Bertozzi, A. L., & Brantingham, P. J. (2013). Community detection using spectral clustering on sparse geosocial data. *SIAM Journal on Applied Mathematics*, *73*(1), 67–83.

van Meter, K. M. (2002). Terrorists/liberators: Researching and dealing with adversary social networks. *Connections*, *24*(3), 66–78.

Verma, T., & Pearl, J. (1988). Causal networks: Semantics and expressiveness. *Proceedings of the Fourth International Conference on Uncertainty in Artificial Intelligence*, 352–359.

Wang, T., Rudin, C., Wagner, D., & Sevieri, R. (2013). Learning to detect patterns of crime. *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, *8190*, 515–530.

Warr, R. L., & Woodfield, T. B. (2019). Bayesian nonparametric estimation of first passage distributions in semi-Markov processes. *Applied Stochastic Models in Business and Industry*, *36*(2), 237–250.

Watney, M. (2020). Law enforcement access to end-to-end encrypted social media communications. *Proceedings of the 7th European Conference on Social Media*, 322–329.

Weiss, G. H., & Zelen, M. (1965). A semi-Markov model for clinical trials. *Journal of Applied Probability*, *2*(2), 269–285.

West, D. B. (2001). *Introduction to graph theory* (Vol. 2). Prentice Hall, Upper Saddle River, NJ.

West, M. (2020). Bayesian forecasting of multivariate time series: Scalability, structure uncertainty and decisions. *Annals of the Institute of Statistical Mathematics*, *72*(1), 1–31.

West, M., & Harrison, J. (1997). *Bayesian forecasting and dynamic models*. Springer.

Wilkerson, R. L. (2020). *Customising structure of graphical models* (Doctoral dissertation). The University of Warwick.

Wilkerson, R. L., & Smith, J. Q. (2021). Customised structural elicitation. *Expert Judgement in Risk and Decision Analysis* (pp. 83–113). Springer.

World Health Organization. (2018). Falls. https://www.who.int/news-room/fact-sheets/detail/falls

Xing, E. P., Fu, W., & Song, L. (2010). A state-space mixed membership blockmodel for dynamic network tomography. *Annals of Applied Statistics*, *4*(2), 535–566.

Xu, J., Marshall, B., Kaza, S., & Chen, H. (2004). Analyzing and visualizing criminal network dynamics: A case study. *International Conference on Intelligence and Security Informatics*, 359–377.

Xu, J. J., & Chen, H. (2005). CrimeNet explorer: A framework for criminal network knowledge discovery. *ACM Transactions on Information Systems*, *23*(2), 201–226.

Yang, C. C., Liu, N., & Sageman, M. (2006). Analyzing the terrorist social networks with visualization tools. *International Conference on Intelligence and Security Informatics*, 331–342.

Zhang, H., & Su, J. (2004). Conditional independence trees. *European Conference on Machine Learning*, 513–524.

Zhang, N. L., & Poole, D. (1999). On the role of context-specific independence in probabilistic inference. *Proceedings of the 16th International Joint Conference on Artificial Intelligence*, *2*, 1288–1293.

Zhao, Z. Y., Xie, M., & West, M. (2016). Dynamic dependence networks: Financial time series forecasting and portfolio decisions. *Applied Stochastic Models in Business and Industry*, *32*(3), 311–332.