Business Decision Insight With Causal Bayesian Networks

An investigation into the potential gains in the application of Causal Bayesian and Constraint Satisfaction Problem AI techniques to aid situational awareness and decision making processes for commercial business problems

Mark Christopher Balmer

MSc by Research

University of York

Computer Science

October 2017

Abstract

Causal Bayesian Networks are a widely recognised tool for modelling the uncertainty of a wide range of processes, particularly when the *nature* of how different factors influence each other. The practice of utilising Causal Bayesian Network is now becoming a growing trend for business that want to fully understand the demands imposed on them, and how best to adapt their business in order to be successful. When designing and building a Causal Bayesian Network, it is often necessary to consult with domain experts for information about the shape of the model but also the definition of how the causal factors influence others. The definition of these influences can require the specification of a large volume of probability distributions, even if a lot of evidential data is available for analysis. Whilst the definition of the structure of the model can be a relatively simple task for a domain expert, providing the probability distributions is a much more difficult task. In this thesis I discuss a method whereby, given a model structure, a domain expert can provide simple descriptive meta-data so that a hypothetical probability distribution can be generated for the discrete model variables.

Contents

Acknowledgements

I would like to thank my supervisor Dr James Cussens for all his guidance and patience throughout the production of this thesis. I would also like to thank my employer MooD International for funding this work and for providing me with the time to perform my research.

Finally, I must express my thanks to my wife Rachael and parents for all their support and encouragement throughout my studies and the process of researching and writing this thesis; and to Lola, for giving me the determination to complete this work.

Declaration

I declare that this thesis is a presentation of original work and I am the sole author. This work has not previously been presented for an award at this, or any other, University. All sources are acknowledged as References.

List of Figures

1 Introduction

A business is composed of many moving parts, and the ability to model and track data which flows between them or is generated by them is a significant task. Many businesses simply do not have adequate knowledge in advanced modelling techniques to fundamentally understand how their business works. This is partly due to only gathering data as outputs of the mechanics of the business. For example, frequently measuring financial data, or manufacturing rate data in isolation. Whilst this data can be gathered to a greater or lesser degree of difficulty; it in itself does not help a business analyst to understand the "why" of the data; that is, the reason or cause of a loss in revenue or a rise in profit, for example. Some businesses invest large amounts of resources and time into building complex models of key business areas, most prominently using a popular spreadsheet application to perform the calculations; some businesses employ Data Scientists to utilise numerous third party data analysis tools to interrogate data.

The trends of "Big Data", "Business Intelligence" and "Business Analytics" tend to be strictly focused on performing Business Process analysis and dimensioning of current and past data to gain an awareness of what the current state of a business is. Invariably these techniques involve some kind of Extract Transform and Load activity (ETL) to gather large volumes of data from numerous sources; then transforming this data into a usable state, and finally loading the data into a centralised repository for further analysis. Whilst these well proven and commonly used techniques can provide vital information to a business in understanding its current situation, they do not help an analyst to understand **why** the process and figures are in their current state. This understanding allows a business to identify key elements, which help model future operative performance.

As identified by industry business analyst Gartner Group, a gap in the market exists which they have described as "Strategy To Execution" [1]. They describe this gap as not only the ability to model business processes, but also the ability to go beyond this into aligning changes to the model, and to measure the changes against business outcomes.

Gathering data is the critical starting point for a business to enable it to begin to understand its business operations. However, this type of data is observational: it is simply a measure of a process at a particular point in time. Whilst this data can be cross referenced, dimensioned and aggregated to give interesting insight into a business; it fundamentally fails to provide insight into how these processes interact, the causal relationships that exist between them and how sensitive they are to change.

Businesses are changing and adapting all the time in response to both external and internal pressures and opportunities, this process of change and adaptation is critical to a business's success. Very rarely can a business simply remain static and hope to continue to be successful. The ability to change and adapt must begin with an understanding of the current situation, but also equally critical is having an understanding of how processes work and their inter-dependencies.

Causal reasoning and the application of Causal Bayesian Networks within businesses, particularly outside of academia or gaming industries, is a relatively rare practice, limited only to research projects within these industries. However, the application of these models can be extremely useful to a business to help them understand the workings of their business operations in a more holistic manner. The employment of Bayesian Networks in academic use is far reaching given the levels of research actively being undertaken to what is still an emerging area. The gaming industry is a natural fit for this type of technology due to the need to understand the player-base and respond accordingly with targeted offers. Therefore, it is easy to see the opportunities to use Causal Bayesian Networks in these industries, however the aim of this thesis is to highlight how these techniques can be applied to everyday business issues, which more often than not only gathers output or response data from sources such as customer feedback or equipment. In this sense the business data is observational in that it does not necessarily help in explaining *why* the gathered data is the value it is.

Should a business decide to use Causal Bayesian Networks, they can be faced with numerous challenges in the identification and construction of the models, and the terminology that is used. As many businesses simply do not gather the types and volumes of data necessary to facilitate the generation of Causal Bayesian Networks from these sources, it is highly likely that the use of domain expert knowledge is required. This, however, presents a problem when taking into account the demands and complexities involved in quantifying the model with what can be large amounts of probability numbers. Assessing the level of uncertainty for how key components of a modelled process in the form of a probability range by a domain expert or experts, can be extremely prone to errors and biases. This necessary activity in the modelling of a business's processes needs to be rationalised and disambiguated, in order to assist the domain experts to specify pragmatic baseline uncertainty assessments.

2 Aim of the thesis

In this thesis, I investigate the application of Causal Bayesian Networks and Constraint Satisfaction Problems to provide insight into business decisions and the generation of probability distributions respectively. The application of these technologies allows a deeper understanding of a business's key components and concepts and also what factors are driving them. The existing business data landscape is composed of a graph structure of inter-connected key business concepts; the relationships between these concepts are expressed as associative, that is, one concept has an association relationship with other business concepts. The addition of AI technologies into this data model, allows for a business analyst to add further business concepts to the model. These can comprise of outside factors, behavioural factors or key influencing factors for which the business has no associated data recorded.

Once a Causal Bayesian Network has been introduced into a business's data landscape, key variables of the causal model can be evidenced to any observational data that is relevant within the business data model. In the case study, the causal model is constructed around the generalised behaviour of a fictitious bank with respect to mortgage application demands and customer satisfaction. To understand the cause and effect of various influencing factors for specific branches, contextual observations for specific model factors can be applied. For example, given the area of a specific bank branch, we can contextually observe the unemployment rate for the area, and report on the model results.

For the design of these Causal Bayesian Networks, I feel that in the current state of data gathering and analysis practices by most businesses, the dependence on domain experts for the definition of the causal models variables, and their influence probability values is still at the forefront. These domain experts are typically key personnel within the business structure but are not necessarily data analysts and almost certainly not statisticians; therefore, it is critical that the domain experts should be able to specify these probabilities in a more natural way, such as using keywords which are based around a more verbal expression. These verbal expressions, along with the model structure and a relatively small amount of meta-data can then be used to generate constraints and ultimately generate a hypothesis about the probabilistic influences of the variables in the model.

To establish numerical values for a set of verbal expressions, it will involve an experiment via the use of a targeted survey to a select group. The survey will be centred around calibrating the numerical value of a specific set of probabilistic words; in which a scenario describing an example of the degree of the word is specified, and the group must indicate a numerical estimate of its magnitude given this situation. It is my hope that a range of descriptive words or phrases will emerge that a subject matter expert might use as an alternative to specifying a potentially large volume of numerical values during the construction of a Causal Bayesian Network. Once these probability descriptive words have been calibrated, a Domain Expert can use them to describe the degree to which a set of model variables affects another. The aim of using these words is that the Domain Expert has only to specify a minimal amount of meta-data so that a system of constraints can be generated and subsequently a range of probability values can be generated from these constraints using traditional Constraint Satisfaction Problem techniques.

To summarise, I propose the integration of key AI technologies into a single coherent causal modelling platform, which can be applied to a business's data landscape; this results in a platform which can be utilised by business analysts and domain experts to gain deeper insights into their businesses to effect change. Subsequently, I propose an uncertainty elicitation method to quantify a Causal Bayesian Network with probability values, via the use of natural language keywords and Constraint Satisfaction Problem solving techniques.

2.1 Research Direction

The research is driven by the requirement to complete a conditional probability distribution by a domain expert in the absence of data. Current available options are for the domain expert to manually enter the probabilities, which can very easily result in an intractable exercise given the size these distributions can be, due to the size growing exponentially with the number of parent variables. A basic option can simply be to restrict the number of parents and their discrete states, however this kind of restriction would be too inflexible.

An option could simply be to utilise qualitative Bayesian networks [2] for the task, whereby the stochastic direction of the influences are captured. The domain experts are required to specify qualitative signs, positive, negative, no change or ambiguous, for each influence in the model. These influence signs are specified from statements such as "as the costs increase, then the profits decrease", which would indicate a negative signed influence. For the domain experts being able encode their knowledge in this manner requires significantly less effort than the quantitative method. However, considering these qualitative indicators of influence direction, they do not allow for an indication of strength and are modelled at a coarse level of detail, as such qualitative modelling can often lead to uninformative results. For some domains this level of detail may be specific enough, however in the domain of business decisions, more detail is required.

Current available third party Bayesian network software, i.e., Hugin, BayesiaLab, have options for the user to specify an expression that completes the distributions. These expressions comprise a standard set of mathematical functions, comparison functions and if-then-else type operators. These expressions would still need to be specified for each value in the conditional probability distribution, and can have similar intractability as manually specifying the values.

There is a well known method which can reduce the number of probabilities that need to be manually specified in the Noisy-OR [3] model, which is a generalisation of the logical OR. This method can compute the values required for the conditional probability distribution from a set of distributions, elicited from the expert. The problem with this model is the assumption that parents act independently on the child variable; also the Noisy-OR method works best when the variables have binary states. In real world business cases, the Bayesian networks being created, the parent variable will most likely not act independently, and will mostly possess more complex states than binary.

Given these current options, and the fact that they fall short of a practical solution in the real world, this research is focused on a method of generating the conditional probability distribution given a simple set of configuration data by a domain expert. Once this configuration is specified, a conditional probability distribution has a series of constraints generated, each value will have one or more constraint generated. Then a constraint solver will produce a candidate distribution for the domain expert to assess for validity against their expectation.

3 Outline of the thesis

The thesis is composed of two parts, which correspond to the two objectives mentioned above. In the first part I describe the mechanics of the AI technologies that are used in Causal Bayesian Networks, and conditional probability distribution value generation using a constraint problem. Also explored are the heuristics and biases that can affect probability assessments made by domain experts.

In the second part of the thesis, a case study is described which is centred on a fictitious bank and its current problems with customer satisfaction and mortgage application demands. The business problem is described, along with a proposed Causal Bayesian Network and discrete model variables conditional probability values which have been generated.

In appendix A the survey analysis data and probability generation investigation analysis data is catalogued. The thesis is concluded with a summary of presented results and some directions for further research areas.

All values detailed in this document are generated by Hugin in the case of the Bayesian network, and Choco Solver [4] in the case of constraint solving. Source code and case study data used in this thesis can be found on GitHub at `https://github.com/mcb539/CPTGen`.

Part I

4 Causal Bayesian Networks

4.1 Probability Theory

In a mathematical model such as a Bayesian Network , whose aim is to encode the influence behaviour between connected variables, it is necessary to cast the results in the language of uncertainty, in which we use probabilities.

Using a language of probabilities is a more common practice in everyday life than we might recognise. How many times during the day do you hear or use phrases like 'It's clouding over, I think it's likely it will rain'. This is still a probabilistic statement even though it isn't stated more like 'It's clouding over, I think there is a 73% of rain'? it's the expression of uncertainty that is important.

When expressing a probability of events occurring, it's based on the principle that there is at least a basic understanding of the majority space in which the event can occur. For example, returning to the above statement, we can assign a probabilistic description to the event of it raining out of the total event space of available weather conditions. Considering a smaller event such as the roll of a dice, then the space of possible outcomes or *outcome space* can be expressed as being one value from the range of values 1 to 6.

This is a basic principle in Bayesian Network modelling, let Ω be the outcome space for a standard dice roll, $\Omega = \{1, 2, 3, 4, 5, 6\}$. Given this outcome space, there is a set of measurable events S to which probabilities can be assigned. In the dice example, the event $\{6\}$ represent the event of the dice roll resulting in a 6, and event $\{2, 4, 6\}$ represents the event of an even numbered dice roll. Let $x \in S$ be a subset of Ω.

Definition 4.1. Event Space

An event space must satisfy the following three principles:

- It contains the *empty event* $\emptyset$, and the *trivial event* Ω.
- It is closed under union. If $x, y \in S$, then so is $x \cup y$.
- It is closed under complementation. If $x \in S$, then so is $\Omega - x$

Definition 4.2. Probability Distribution

A probability distribution P over (Ω, S) is a mapping from the events in S to real values that satisfy the following three conditions [5]:

- $P(x) \geq 0$ for all $x \in S$
- $P(\Omega) = 1$
- If $x, y \in S$ and $x \cap y = 0$, then $P(x \cup y) = P(x) + P(y)$.

Definition 4.3. Conditional Probability

A conditional probability describes the amount of certainty relating to a variable x given what we know about the certainty of another given variable y. This is expressed as $P(x|y)$.

P is a joint probability distribution on a set of variables U, and $X, Y \subseteq U$. Any combination of x for X, and y for Y, where $P(y) > 0$ the conditional probability of x given y is expressed as:

$$P(x|y) = \frac{P(y \cap x)}{P(y)}$$

Definition 4.4. Chain Rule

With reference to the definition of Conditional Probability , it's clear that $P(x \cap y) = P(x|y)P(y)$, this is the fundamental expression of the chain rule. If $x_1, ..., x_i$ represents a sequence of events, then total distribution can be expressed as:

$$P(x_1 \cap ... \cap x_i) = P(x_1) \cdot P(x_2|x_1) \cdot ... \cdot P(x_i|x_1 \cap ... \cap x_{i-1})$$

The chain rule describes a method to determine the joint probability of a sequence of events by the probability of the first event, then the probability of the second event given what we know about the first event, and so on.

Definition 4.5. Bayes' Rule

Bayes' Rule describes a method of computing the inverse of a conditional probability given a conditional probability; for example, given a conditional probability $P(x|y)$, using Bayes' rule it is possible to determine the estimated probability of Y given the information available from the conditional probability about x and the prior probability information about x and y.

$$P(x|y) = \frac{P(y|x) \cdot P(x)}{P(y)}$$

for all combinations of values x with $P(y) > 0$

Definition 4.6. Random Variables

When discussing events as sets of outcomes, it is better to consider the attributes of the outcome. For example, given a Person, then the person's attributes could be *height*, *weight* and *age*, and these could be used to reason about a person's health. These attributes can then have probabilities assigned to them, and then events can be considered such as *age > 60, height is 6 foot and weight is 18 stone.* It is these attributes and their values for different outcomes which are defined a random variables.

A random variable is defined by a function which associates to an outcome space Ω a value. For example, a random variable *gender* is defined by a function f_{gender} which maps each person in Ω to their gender, M or F.

The values that can be assigned to random variables can be one of two types, either *discrete* or *continuous.* That is, the value can be categorical or integer / real values respectively. For this thesis random variables are either discrete or real-valued continuous, and are denoted by uppercase letter X, Y, Z.

Definition 4.7. Marginal Distribution

Given a random variable X, the distribution by which the events could occur on X is referred to as the marginal distribution over X, and is expressed as $P(X)$. For example, the marginal distribution over a random variable *weather* given the event space of $\Omega = \{sunny, rain\}$, the marginal distribution could be $P(weather = sunny) = 0.71$ and $P(weather = rain) = 0.29$. Marginal distributions must adhere to the conditions outlined in definition 4.1.

Definition 4.8. Joint Distribution

Often we will be required to ask a question which involves the values from several random variables, for example given the random variables *weather* W and *region* R, we might be interested in the event $weather = sunny$ and $region = north$. The marginal distribution of the random variable *region* is defined as $P(region = north) = 0.70$ and $P(region = south) = 0.30$.

When an event contains several random variables the concept of a *joint distribution* must be employed over the random variables. A joint distribution is a distribution which assigns probabilities to a set of events with respect to the random variables.

The joint distribution must be consistent with the marginal distribution of the random variables, given that $P(x) = \sum_y P(x, y)$. This is shown in table 1 assuming the set of events is $\{region, weather\}$, by summing the columns of joint probabilities for *sunny* we can arrive back at the marginal probability, and likewise for all other events. In this example there are sixteen atomic outcomes given the two random variables.

		weather		
		sunny	rain	
region	north	0.652	0.048	0.70
	south	0.058	0.242	0.30
		0.71	0.29	1.00

Table 1: Joint distribution example for $P(weather, region)$

Definition 4.9. Conditional Probability Distribution

Given a joint distribution we can alternatively express in a more natural way with respect to the chain rule as detailed in definition 4.1. For example, given the joint distribution in table 1, the joint distribution can be expressed as $P(W, R) = P(R)P(W|R)$. Therefore it is more desirable to express the joint distribution by using each random variable's *conditional probability distribution (CPD)* values. Given this example the CPD of *region* represent a *prior distribution* and *weather* is a *conditional probability distribution*

region	
north	south
0.70	0.30

Table 2: CPD for the region variable

		weather	
		sunny	rain
region	north	0.931	0.069
	south	0.193	0.807

Table 3: CPD for the weather variable

Definition 4.10. Marginalisation

When we have a joint distribution on two variables X and Y, the marginal probability is described as a subset of the joint distribution by summing over the other variables to which the value is observed.

P is a joint probability distribution on a set of variables U, $Y \in U$ is a variable with values $y_i, i = 1, ..., n$; let $X \subseteq U$, then the marginal probability of X is

$$P(X) = \sum_{i=1}^{n} P(X \wedge Y_i)$$

for all combinations of values x for X define a joint probability distribution on X.

Definition 4.11. Independence

Given a distribution $P(X|Y)$, the variable X is dependent on changes in belief of Y. However, given two random variables X and Y, if $P(X|Y) = P(X)$, then the variables X and Y are said to be independent; any changes in belief about Y has no effect upon the belief of X. The notation for this independence is given as $(X \perp Y)$.

A distribution P satisfies $(X \perp Y)$ if and only if $P(X \cap Y) = P(X)P(Y)$ [5]

Proof. $P(X|Y) = \frac{P(X \cap Y)}{P(Y)} = \frac{P(X)P(Y)}{P(Y)} = P(X)$ □

Given this definition its clear that independence is symmetrical.

Definition 4.12. Conditional Independence

Conditional independence describes the notion that given information about an event, it does not provide any other information about other events. Two events X and Y are conditionally independent given a third event Z, if the both events X and Y are conditionally independent given a third event Z.

$$(X \perp Y|Z), \text{ if and only if } P(X \cap Y|Z) = P(X|Z)P(Y|Z) \text{ [5]}$$

Events can become conditionally dependent given a third event, however it is highly dependent on the nature of the third event.

For example, if two separate dice are rolled, these two events are independent from each other; each dice roll does not influence the other dice roll. A third event of the sun shining is also independent of the dice rolls. However, if the third event is the probability that the sum of the dice rolls is even, then the two dice roll events become conditionally dependent given the third event, as knowing the first dice roll value and the probability of the third event infers belief about the second dice roll [6].

4.2 Graphs

Throughout this document the techniques used are primarily based upon Causal Bayesian Networks (see section 4.5), which in turn use graphs as their base construct.

4.2.1 Vertices and Edges

A graph structure G consists of a set of vertices and a set of edges, which can be either directed, undirected or in some special circumstances bi-directed.

In this thesis a vertex is represented by V and a set of vertices $\nu = \{V_1, ..., V_n\}$. A pair of vertices V_i and V_j can be connected by a directed edge, represented as $V_i \rightarrow V_j$, or as an undirected edge, represented as $V_i - V_j$. Therefore the set of edges E in the graph is a set of vertex pairs.

Furthermore, a variable is synonymous with a graph vertex. For the purposes of this thesis, only directed graphs are considered. However, during the graph transformation process of constructing a clique graph, it is necessary to describe the usage of undirected edges.

When there is directed edge $V_i \rightarrow V_j \in E$, we define V_i as the *parent* of V_j, and that V_j is the *child* of V_i. In this thesis parents are represented as Pa_V, which denotes the parent vertices of V. Child vertices are represented as Ch_V to denote the child vertices of X.

Directed graphs encode independence between the variables in the graph, therefore building on the independence definitions in section 4.1, it will be necessary to describe the usage of a bi-directed edge.

Definition 4.13. Directed Acyclic Graph

A directed acyclic graph is the primary graphical representation for a Bayesian Network. If all edges in a graph are directed, indicated by a single arrowhead, and the graph contains no cycles, then the graph can be considered a directed acyclic graph .

Let G be a directed graph with a pair $G = (\nu, E)$, where ν is a finite set of vertices and E is a set of ordered pairs of vertices $V_i \rightarrow V_i \in E$, as edges.

Definition 4.14. Markov Blanket

The Markov Blanket [3] of a graph vertex describes a vertex and edge set which shield the vertex from the remainder of the graph. In this the Markov Blanket consists of the vertex parents, its children and its children's parents.

Let G be a DAG. Let V_i, V_j be vertices in G. Vertex V_j is a parent of vertex V_i if $V_j \rightarrow V_i \in E$; vertex V_i is a child of V_j.

The set $Pa(V_i) \cup Ch(V_i) \cup Pa(Ch(V_i))$ is the Markov Blanket of vertex V_i.

Definition 4.15. Path

A path is an ordered alternating sequence of vertices and edges which describes the route from vertex V_i to V_k.

Let $G(\nu, E)$ be a DAG. Let $\nu = \{V_0, ..., V_k\}, k \geq 1$, be a set of vertices in G. A path p from vertex V_0 to V_k in G is defined as the sequence $V_0, E_1, V_1, ..., E_k, V_k$ of vertices and edges $E_i \in E, i = 1, ..., k$. Each edge $E_i \equiv V_{i-1} \rightarrow V_i$ or $E_i \equiv V_i \rightarrow V_{i-1}$ for every ordered pair of vertices V_{i-1}, V_i in the sequence. k is the length of the path p. If every edge between the vertices in a path has an arrow pointing in the same direction from the first to the second vertex, then the path is a *directed* path.

Any concatenation of two paths in G, results in a path in G.

Definition 4.16. Cycle

A cycle in G is a directed path p with a length of one or more, where the head and tail of the path are both V_0. It is assumed in this thesis that any graphs depicted are directed acyclic graphs , and therefore will not contain any cycles, unless otherwise stated.

4.3 Bayesian Networks

The concepts of probability theory and graph theory come together as graphical models in the use of Bayesian Networks. The variables are represented as vertices in the graph, and are also commonly referred to as nodes; the probabilistic relationship between variables is represented by a graph edge, again more commonly referred to as an arc. Any conditional independence in the structure is represented by the lack of an arc between variables. By representing the probability distribution as a directed acyclic graph, the flow on probabilistic influence can be computed exactly.

Using a directed acyclic graph as a structure to model the framework for a joint probability distribution, the graph nodes represent a probability variable and so for the remainder of this document there will be no explicit distinction made between the two concept of variable and node. Conditional independence is captured by a graph edge in the DAG, via these relationships in the graph structure the variable independence is described.

A Bayesian network graph structure is comprised of three basic node structures: chain, fork and collider. From these building blocks it is possible to create a fully connected Bayesian network which adequately describes a joint probability distribution.

Definition 4.17. Chain

Let $G(\nu, E)$ be a directed acyclic graph. Let V_i, V_m and V_j be vertices in ν. If edges exist such that $V_i \rightarrow V_m \in E$ and $V_m \rightarrow V_j \in E$ then the vertex set $\{V_i, V_m, V_j\}$ is said to be a chain.

Definition 4.18. Fork

Let $G(\nu, E)$ be a directed acyclic graph. Let V_i, V_m and V_j be vertices in G. If edges exist such that $V_i \leftarrow V_m \in E$ and $V_m \rightarrow V_j \in E$ then the vertex set $\{V_i, V_m, V_j\}$ is said to be a fork.

Definition 4.19. Collider

Let $G(\nu, E)$ be a directed acyclic graph. Let V_i, V_m and V_j be vertices in G. If edges exist such that $V_i \rightarrow V_m \in E$ and $V_m \leftarrow V_j \in E$ then the vertex set $\{V_i, V_m, V_j\}$ is said to be a collider.

Definition 4.20. Independence Map

Let $G(\nu, E)$ be a directed acyclic graph , and let P be a joint distribution on ν. G is an independence map, or I-map for P if

$$\langle X|Z|Y\rangle_G \Rightarrow P(X|YZ) = P(X|Z)$$

for all sets of variables $X, Y, Z \subseteq \nu$.

An independence map is a directed acyclic graph which encodes the independence of variables: any variables that do not have an arc between them are independent of each other is the joint probability distribution. In order for Bayesian network to accurately model a probability distribution, each variable is conditionally independent of its non-descendants given the values of all its parent variables.

In order to determine the correct independence between variables within the graph structure, the concept of d-separation can be used. D-separation is a criteria for inferring whether two sets of variables which are connected by a path are conditionally independent from each other, given a third set of variables.

Definition 4.21. d-Separation

A path p is said to be blocked, by a set of nodes Z if and only if

- p contains a chain $i \rightarrow m \rightarrow j$ or a fork $i \leftarrow m \rightarrow j$ such that the middle node m is in Z
- p contains a collider $i \rightarrow m \leftarrow j$ such that the middle node m is not in Z and such that no descendant of m is in Z.

A set Z is said to d-Separate X from Y if and only if Z blocks every path from a node in X to a node in Y [7]

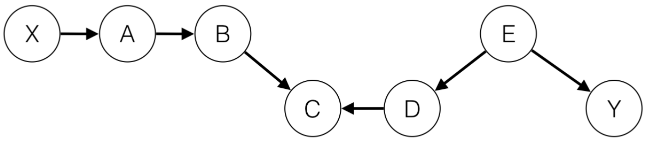


Figure 1: d-separation given a DAG which contains no conditioning set Z

The DAG illustrated in figure 1 contains no conditioned vertices, however X and Y are d-separated due to there being a single collider vertex C.

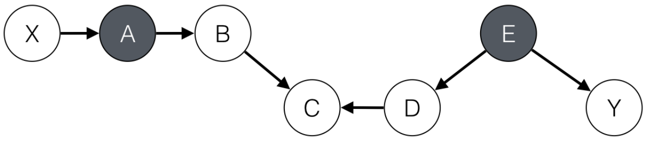


Figure 2: d-separation given a DAG which contains a conditioning set Z on A and E

The DAG illustrated in figure 2 contains a conditioning set $Z = \{A, E\}$ (indicated by the dark circles). The set Z does not condition on the collider vertex C, therefore X and Y are still d-separated. However, given the set Z, X and B are now d-separated, and so are D and Y. In this example only B and C and D and C are d-connected.

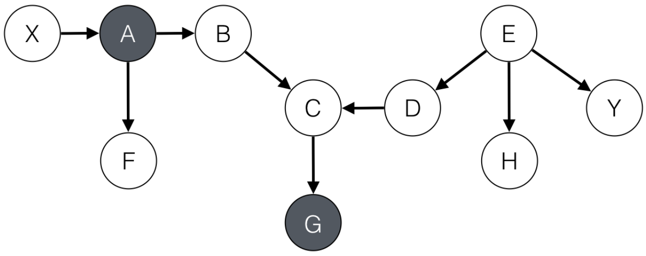


Figure 3: d-separation given a DAG which contains a conditioning set Z on A and G

The DAG illustrated in figure 3 contains a conditioning set $Z = \{A, G\}$ (indicated by the dark circles). The vertices B and Y are d-connected by Z, this is because the collider vertex C has a descendant vertex G in Z which *unblocks* the path from B to Y. The path from X to D is still d-separated due to the vertex A being in Z.

4.4 Inference in Bayesian Networks

The graphical structure of a Bayesian Network provides a good mechanism for an analyst to understand the relationships between concepts, which are represented as variables within the network; furthermore, the graphical structure affords the evaluation of the graph's structural properties, such as independence, in a much more digestible way. The other major benefit of a Bayesian Network is the ability to use the structure for inference, so that queries on distribution structure can be answered. The process of inference is to make observations about the concept being modelled, and then calculate the posterior probability of some variables.

Given a directed acyclic graph structure, which for example a domain expert(s) have created, in order to compute inference on this structure there are several graph transformations and algorithms that need to be performed; these transformations and algorithms are covered in this section.

Firstly, in order to be able to perform the posterior probability calculations, the graph structure must be transformed into a tree structure of cliques, called a clique graph; this transformation process involves several algorithms and is known to be an NP-Hard problem to produce an optimal tree structure. Secondly, once a clique graph has been created exact inference can be calculated for all variables in the network.

This thesis explores using Bayesian Networks which have a mixture of both discrete and continuous variables, commonly known as hybrid networks. Consequently, the transformation and calculation algorithms used to perform exact inference on hybrid networks have subtle and important differences to those used if the network were composed solely of discrete variables. The case study for this thesis depends on being able to calculate exact continuous values for monetary amounts, and as such using a discrete only model poses flexibility and accuracy problems, due to the necessity to discretize the variable values; in some circumstances this discretization may be desirable, as it represents an approximate value, however this concept is not explored in this document.

4.4.1 Moralization

The initial step of decomposing a Bayesian Network graph structure into a clique graph involves the process of moralizing the graph structure. Moralization casts the directed acyclic graph structure into an undirected graph structure, whereby the parent variables $PA(X)$ of variable X have an undirected edge between them if an edge does not exist. The moral graph G^m from the original graph G is moral if for each pair of variables V_x, V_y that share a child variable, there is an edge between V_x and V_y.

Definition 4.22. Moralization

Let $G(\nu, E)$ be a directed acyclic graph. Let U be the undirected graph of G. Let G^m be a moral graph of G over U if either:

- an edge exists such that $V_x - V_y$ or $V_y - V_x \in E$
- V_x and V_y are both parents on of the same vertex

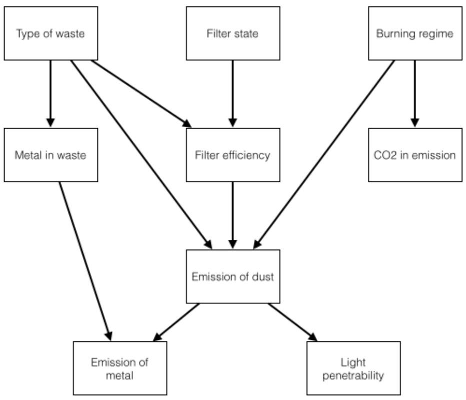


Figure 4: Original DAG

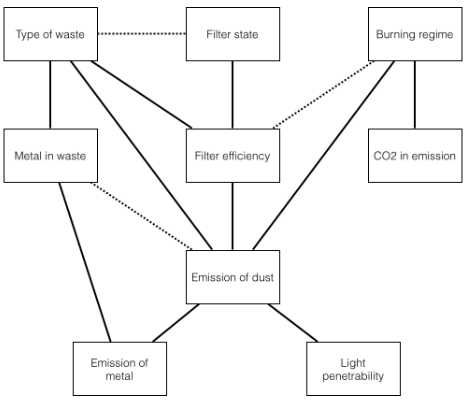


Figure 5: Moralized graph, dashed lines indicate moral edges added

Definition 4.23. Chordal Graph

Let $G(\nu, E)$ be an undirected graph, and G^m be a moral graph of G. A cycle σ exists in G^m with the cycle being in the sequence $\{v_0, v_1, ...v_n\}$ with $v_0 = v_n$. A *chord* of cycle σ is a pair of vertices in ν (v_i, v_j) which are non-neighbouring vertices in σ such that an edge exists between them $v_i - v_j$.

The undirected graph G is a chordal graph if any cycle in the graph which has a cycle length of 4 or more vertices, contains a chord which ensures that every cycle in the graph has at most 3 vertices.

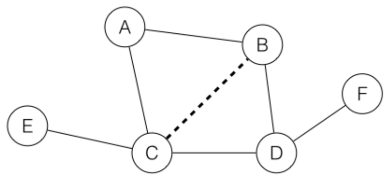


Figure 6: A chordal graph, the chord detailed by the dashed edge $B - C$ triangulates the cycle $A - B - D - C - A$

The addition of chords to eliminate large cycles is through an algorithm known as *triangulation*; the algorithm requires an ordering of vertices to systematically eliminate from the undirected graph. At each elimination the remaining undirected graph is inspected for cycles of length ≥ 4 and a chord (also commonly referred to a fill-in edge) is added.

4.4.2 Junction Tree

A Junction Tree transformation of a DAG is designed to represent and enable computations on the joint distribution of the original graph representation. The transformation process is performed using the undirected moral graph and ensures that the factorisation of variables within the graph structure remain intact; any loops within the moral graph are eliminated via a process known as *triangulation*, which results in an undirected graph structure known as a *chordal graph*.

A Junction Tree is defined by a specific structural constraint referred to as the Running Intersection Property:

Definition 4.24. Running Intersection Property

Let T be a tree graph structure, and let C be a collection of variable subsets of the set of vertices V. T is a junction tree if any intersection $C_1 \cap C_2$ of a pair of sets C_1, C_2 in C is contained in every variable on the unique path between C_1 and C_2.

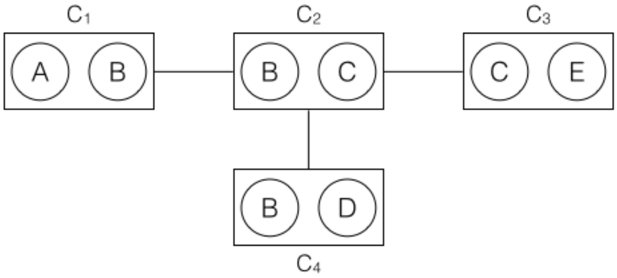


Figure 7: A junction tree with the running intersection property in C_1 and C_2, C_1 and C_4 , C_2 and C_3, C_2 and C_4.

4.4.3 Variable Elimination Ordering

Algorithm 1: Greedy search for constructing an elimination ordering

M // a moralized graph over ν

initialise all variables in ν as unmarked

for $k = 1 ... |\nu|$ **do**

  select an unmarked variable $V \in \nu$ with the minimum edges needed to be added if eliminated

  $\pi(V) \leftarrow k$

  Add new undirected edges in M between all neighbours of V

  mark V

end

return π;

At the point of assigning a cost value to the unmarked node, the algorithm evaluates a cost based upon the amount of additional edges that would have to be added to the graph in order to produce a clique, the selection will be based upon a minimum of fill-in edges. This minimum fill-in cost based heuristic ensures that the complexity to create the clique graph is kept to a minimum. Once the algorithm is complete the variables π will contain an ordering of variables from ν in reverse order by which the triangulation process can be performed. Algorithm 1 is a good all-purpose variable elimination algorithm to use for what is an NP-Hard problem; the aim of generating an efficient variable elimination ordering is so that during triangulation and clique building for the junction tree, the created cliques are kept as small as possible. It is desirable to keep the junction tree cliques to a small size so that computation within the cliques is also kept to a small size, and as such the speed of calculation is kept low.

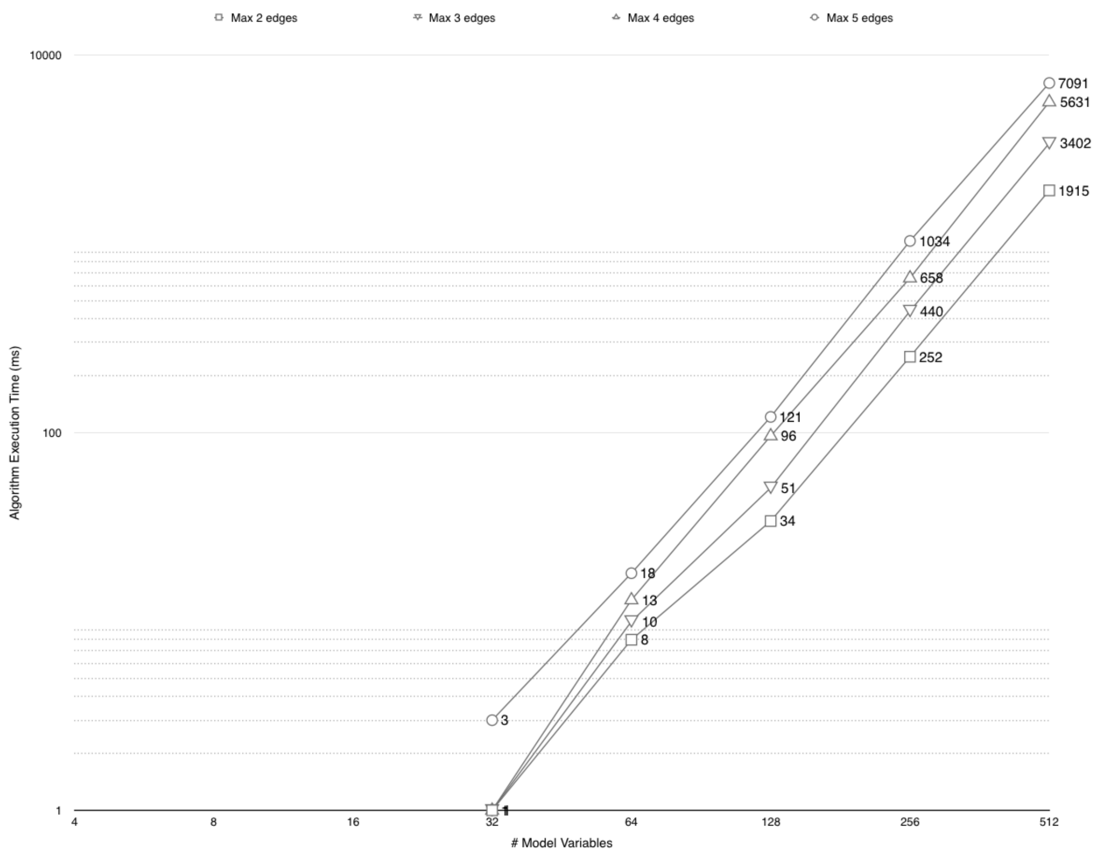


Figure 8: Variable Elimination Ordering Algorithm Performance

4.4.4 Triangulation and Clique Building

If the graph is not yet chordal, then it can be made chordal via the triangulation algorithm by the additional of fill-in edges. Given an ordering of the graph variables, triangulation occurs by analysing each variable in turn of the reverse order of the elimination ordering; each variable under inspection is joined to each neighbour that appears earlier in the ordering, but are not already joined to the variable under inspection.

Once each neighbour is joined to the variable that is under inspection, this cluster of graph variables is identified as a *clique*. It is possible that the set of cliques which are generated from the chordal graph contains cliques which are proper subsets of other cliques. These subsets can be removed from the candidate set of cliques, as they are subsumed by the super-set cliques.

Definition 4.25. Clique

Let C_i be a subset of vertices ν from the graph $G(\nu, E)$, such that the every pair of vertices in C_i is joined by an edge. This subset is referred to as being *complete*. A subset of vertices which forms a subgraph, whereby it cannot be extended by adding adjacent vertices, is known as a complete subgraph which is *maximal*. A subgraph which is maximal is referred to as a clique.

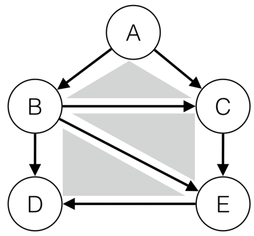


Figure 9: A directed graph, with the cliques {A,B,C}, {B,C,E} and {B,D,E}

As previously mentioned, the variable elimination ordering is critical to the production of optimally sized cliques, this optimisation problem of finding triangulations for undirected graph is an NP-Hard problem.

4.4.5 Strong Triangulation

When performing triangulation for hybrid networks, the triangulation algorithm needs to be modified into a strong triangulation. When calculating hybrid causal models, a specific rule exists when marginalizing over cliques which contain both discrete and continuous variables. In these cases we must first marginalize over the continuous variables and then marginalize over the discrete variables. When marginalizing over the discrete variables if all the variables within the clique are discrete, then a strong marginalization can be performed; otherwise a weak marginalization must be performed. In order to exploit this computational behaviour, we must ensure that *strong triangulation* is performed, to facilitate this the concept of a *strong decomposition* is introduced [8]. *Strong decomposition* and *strong triangulation* both operate on a *marked graph*, a *marked graph* is where the variables of the graphs are marked with their type, discrete Δ or continuous Γ.

Definition 4.26. Strong Decomposition

Given a set of vertices $V(A, B, C)$ in an undirected graph G, is said to form a strong decomposition of G if $V = A \cup B \cup C$ and all the following three conditions hold true:

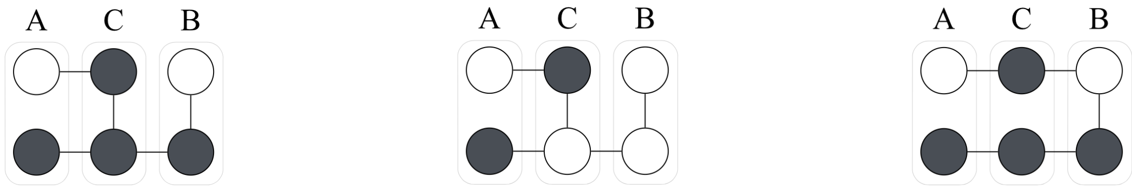
i C separates A from B

ii C is a complete subset of V

iii $C \subseteq \Delta$ or $B \subseteq \Gamma$

Figure 10 shows various examples of the rules for strong decomposition.

Strong triangulation is the same as the standard discrete only network triangulation, except that an additional step of adding edges prior to the triangulation. These additional edges to be added must link non-neighbouring discrete variables, when these variables have



(a) (b) (c)

Figure 10: Strong decomposition. Solid variables are discrete, clear variables are continuous. (a) strong decomposition with $C \subseteq \Delta$. (b) strong decomposition with $B \subseteq \Gamma$. (c) no strong decomposition as the variables in C are not complete.

a path between them which contain any continuous variables; these additional graph edges enforce the strong decomposition rule.

Algorithm 2: Add strong elimination edges

Result: Add Strong Triangulation Edges

foreach *discrete variable d in the moral graph G^m* **do**

  $n \leftarrow$ all non-neighbouring discrete variables of d

  foreach *non-neighbouring variable v in n* **do**

    $p \leftarrow$ count of any shortest paths between d and v which exclusively go through any continuous nodes

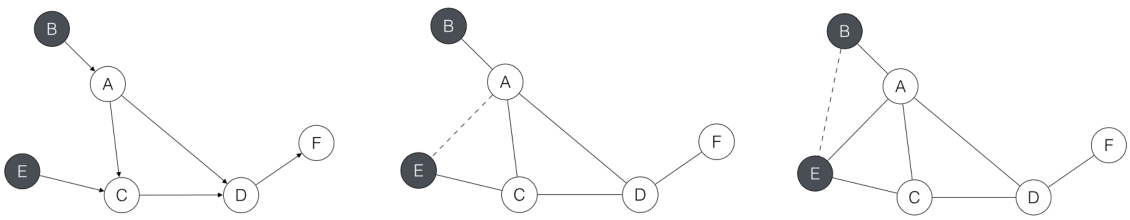
    if $p > 0$ **then**

      add a new undirected edge e between d and v

    end

  end

end

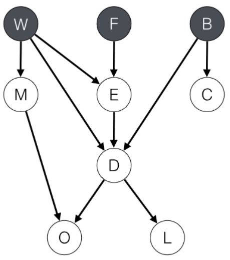


(a) base directed graph (b) moralized (c) strongly triangulated

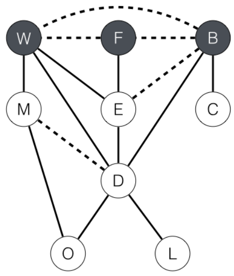
Figure 11: Strong triangulation. Solid variables are discrete, clear variables are continuous.

4.4.6 Clique Trees

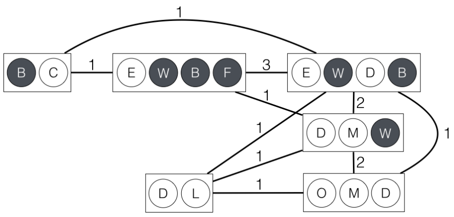
Once the cliques of variables have been generated, it is necessary to join the cliques into a tree structure; that is a structure whereby every pair of cliques has one and only one direct path between them. As previously stated the resulting clique tree must exhibit the *running intersection property*. The method chosen to construction the clique tree that satisfies these properties is an implementation of the Maximum Spanning Tree which is a modification of the Kruskal Minimum Spanning Tree algorithm, the modification is the inversion of the weight attributed to the clique edges so that the algorithm calculated the maximum costing path. In this case, the clique edge weights are the cardinality of intersecting variables given a pair of cliques.



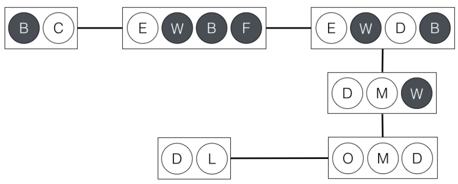
(a) base directed graph



(b) moralised and strongly triangulated



(c) cliques with weighted intersection joins



(d) selected joins from Maximum Spanning Tree algorithm

Figure 12: Strong triangulation. Solid variables are discrete, clear variables are continuous.

The final clique tree in fig 12 (d) exhibits the *running intersection property* for the selected clique edges added to form the clique tree, therefore satisfying definition 4.24.

A *separator* is a structure which contains the message data and the structure of the separator is determined by the intersection of clique variables for each pairing of cliques; Therefore, the number of separators in a clique graph is the same as the number of graph undirected edges between cliques.

Definition 4.27. Clique Separator

Let C_i and C_j be two cliques in the clique tree which are connection by an undirected edge, then $S_{i,j} = C_i \cap C_j$ is a separator between C_i and C_j.

Figure 13 shows the separators for an example clique graph.

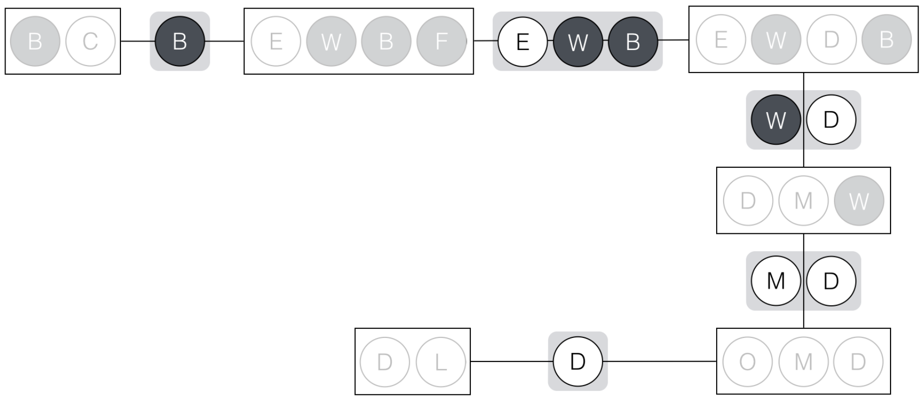


Figure 13: A clique graph with the separator structures highlighted

Given a tree structured graph T whose nodes are maximal cliques $C_1, ..., C_n$, then $W_{<(i,j)}(W_{<(j,i)})$ are all the variables that appear in any clique on the $C_i(C_j)$ side of the edge [5].

Definition 4.28. Clique Tree

A tree structured graph T is a clique tree for an undirected graph H if:

- each node in T corresponds to a clique in H
- each maximal clique in H corresponds to a node in T
- each separator $S_{i,j}$ separates $W_{<(i,j)}$ and $W_{<(j,i)}$ in H

4.4.7 Clique Factors

The sets of random variables in a Bayesian network can be described as a *factor*, that is each variable describes a function to a value, these factors can be used to define the joint probability function, and can be defined as:

Definition 4.29. Factor Let D be a set of random variables, a factor φ is a function from $Val(D)$ to $\mathbb{R}$

Definition 4.30. Joint Probability Density

$$P(x) = \prod_{v \in V} P(x_v | x_{PA(x_v)})$$

Each *factor* in the Bayesian network must be allocated to an appropriate clique in the clique tree.

Definition 4.31. Factor Clique Assignment

Let C be a clique with a variable set V, a factor φ is assigned to one and only one clique C if V is a superset of φ

$$C(V) \supseteq \varphi$$

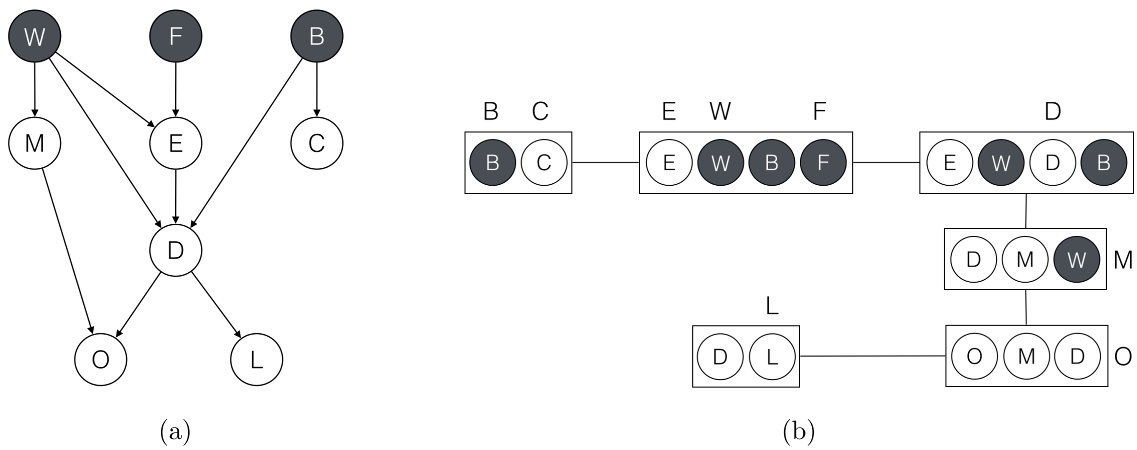


Figure 14: Factor assignment to a clique tree. (a) the original DAG. (b) the clique graph, assigned factors are shown next to the cliques.

4.4.8 Strong Roots

When computing exact inference for hybrid networks on a strong clique tree, it is necessary to initiate the computations from a clique which can be labelled as a *strong root*.

Definition 4.32. Strong Root

In a junction tree, any clique R is a strong root, if any pair of neighbouring cliques A, B with A closer to R than B and satisfies

$$(B \setminus A) \subseteq \Gamma \text{ or } (B \cap A) \subseteq \Delta$$

In order to determine each strong root in the junction tree, in turn each junction tree neighbouring clique separators are inspected against the condition detailed in definition 4.32 in both directions. An intermediate table of results can be generated as shown in table 4, then each junction tree clique is analysed by inspecting each definition entry in the intermediate table to determine if it passes the rule in definition 4.32. The analysis of each clique separator is done by via an inward sequential ordering from the furthest distance clique to the immediate neighbouring cliques from the clique being inspected; an example implementation of this rule is shown in figure 15. If and only if each separator passes the rule in definition 4.32 can the clique being analysed be a strong root.

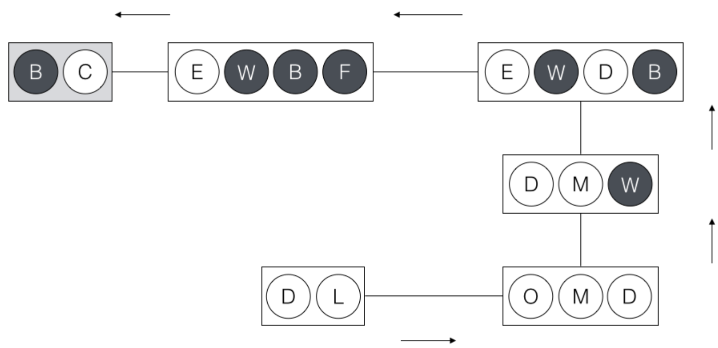


Figure 15: A junction tree showing the *inward* direction check of each clique pairs separator variables, when analysing clique (B, C)

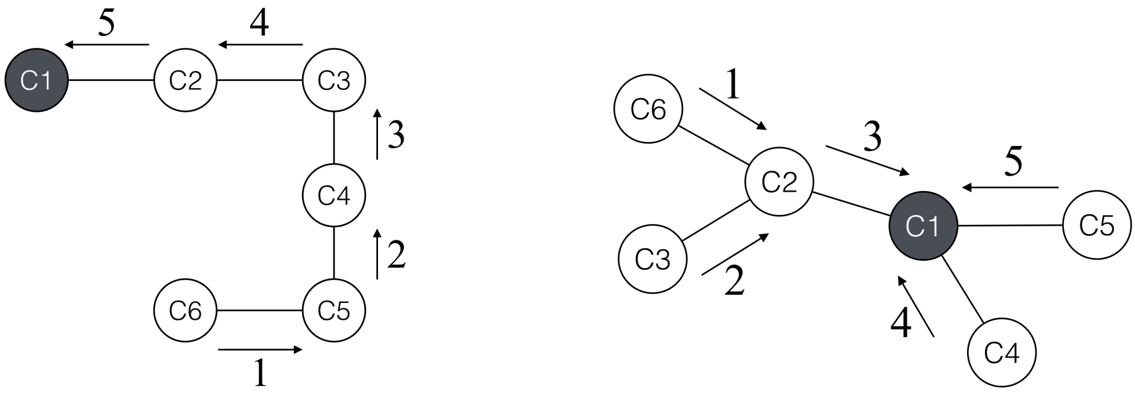
Clique Separator Strong Root Analysis				
B	A	$(B \cap A) \subseteq \Delta$	$(B \setminus A) \subseteq \Gamma$	Passes definition 4.32
BC	EWBF	B	C	Yes
EWBD	BC	B	EWF	**Yes**
EWBF	EWBD	WBE	F	No
EWBD	EWBF	WBE	D	**Yes**
EWBD	DMW	WD	EB	No
DMW	EWBD	WD	M	**Yes**
DMW	OMD	DM	W	No
OMD	DMW	DM	O	**Yes**
OMD	DL	D	OM	Yes
DL	OMD	D	L	**Yes**

Table 4: Junction tree clique separators tested against definition 4.32. Underlined variable letters are discrete nodes. Bold values for yes/no indicate the lookup values for the analysis of clique (B, C) as shown in figure 15.

4.4.9 Message Passing Algorithm

When calculating the causal model on the clique graph, we utilise the *message passing* algorithm which ensures that the calculation dependencies which are encoded into the clique graph via the junction tree algorithm are adhered to. As the algorithm name suggests the *message passing* algorithm involves each clique in the clique graph passing a belief propagation message to its neighbouring cliques after each clique has performed its local calculation. When calculating the algorithm begins at the strong root clique, then a recursive traversal of the graph structure is performed; during the traversal, at each clique a decision is made to continue onto the cliques unvisited neighbours if any exist. Figure 16a shows the graph traversal order from the strong root clique $C1$; in turn the cliques $C2, C3, C4, C5$ are visited, at each of these cliques it is true that an unvisited neighbouring clique exists which is further away from the strong root. Once the clique $C6$ is visited this is no longer true, therefore $C6$ performs its local calculation and passes its belief message, called a *separator message*, on to clique $C5$; this recursive inspection continues and hence $C5$ performs its local calculation including the message from $C6$ and then passes its belief message to clique $C4$ and so on.

This recursive local calculation and message passing continues until all the cliques have been visited and the algorithm returns back to the strong root. Figure 16b shows a more complex tree structure and a potential message passing ordering.



(a) (b)

Figure 16: Example graphs highlighting the possible ordering of collect messages.

As the collect phase of the message passing proceed, belief messages are collected from adjacent clique nodes to form a belief message to pass to the next clique node in the recursion. For example, figure 17 highlights the order of clique processing during the recursion (the ordering numbers are shown next to the cliques). In this example the clique B,C is the strong root and so the recursive collect starts here and ends at the clique D,L, whereby the factor $\varphi(D, L)$ is calculated. As clique D,L is a leaf node, the collect step moves to clique O,M,D which needs the belief about D as part of its local belief calculation. Therefore, given the information about the separator between these two cliques, S_D, we marginalize to the separator to form the belief message required. This message $\delta_{1\to 2}$ is then multiplied by the factor of clique O,M,D $\delta_{1\to 2} = \sum_L \varphi(O, M, D)$. This process continues until the recursive graph traversal settles back at the strong root clique.

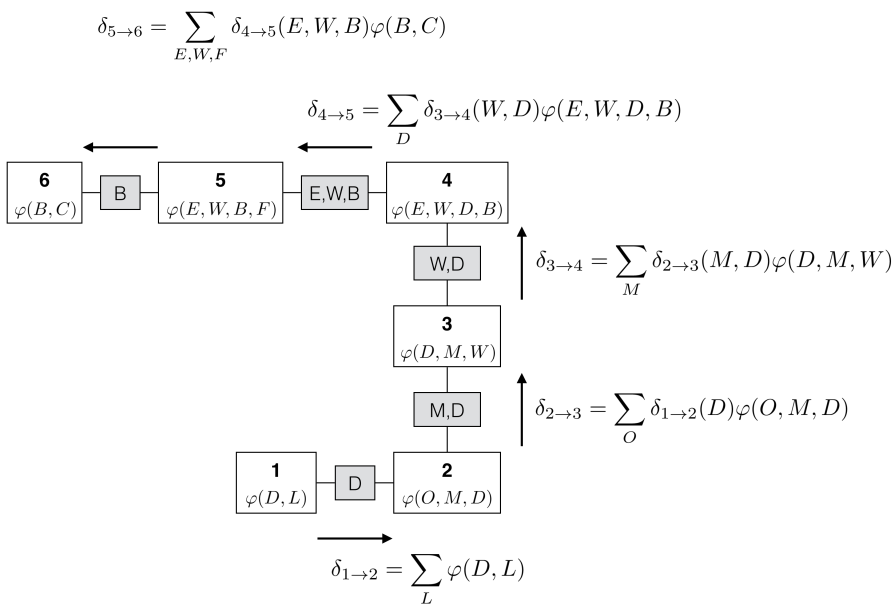


Figure 17: An example graph highlighting the belief message propagation, each numbered clique calculates and passes a message to the next clique.

This is the initial stage of the message passing algorithm, and given the inward flow of belief messages towards the strong root, is named the *collect* stage. Once the collect stage has finished, a second stage is performed which flows belief message outward from the strong root in a reverse manner to that of the collect stage. The distribute stage is necessary so that the strong root clique can pass on to the rest of the clique graph the updated beliefs received during the collect stage.

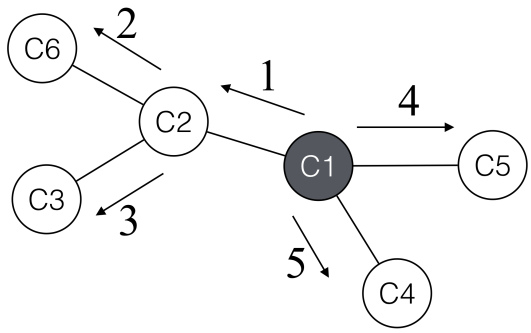


Figure 18: An example graph highlighting a possible ordering of distribute messages.

4.4.10 CG Potential

Performing exact inference on a hybrid Bayesian network introduces additional complexities over inference in a purely discrete Bayesian network. The clique local and message passing calculations must continue to support all of the probability operations we expect: product, division, marginalisation and factor reduction. These operations must also be capable of being performed on both discrete and continuous variables, therefore the structure of a canonical form is introduced. Continuous variables within a hybrid Bayesian network are represented as a conditional Gaussian distribution and discrete variables are represented as a probability distribution across the variables discrete states.

Definition 4.33. Canonical Form

A canonical form is represented by the notation:

$$C(g, h, K)$$

Let C be a canonical form representation. The g property represents the conditional probability of the clique's discrete variable factors. The h property represents the vector of means (μ) of the clique's continuous variable factors. The K property presents the covariance matrix of the clique's continuous variable factors.

Lauritzen and Jensen [9] introduced as their basic computational object a *CG potential*, which in a hybrid Bayesian network we use to represent both the discrete and continuous conditional distributions within the model. For each discrete variable we specify a CG Potential for the conditional probability distribution of the variable given its parents. Given the restriction of a hybrid Bayesian network that no continuous variable can have any discrete child variables, all parents of a discrete variable must be themselves discrete. For each continuous variable the conditional distribution if represented by a normal Gaussian distribution of the type specified in definition 4.34, we must initialise a CG Potential for each configuration of the state space of the discrete parents.

Definition 4.34. Continuous variable Gaussian distribution

$$\phi(V|PA(V)) = \mathcal{N}(\alpha(i) + \beta(i)'z, \gamma(i))$$

For continuous variables, the parent variables $PA(V)$ can be a mixture of both discrete and continuous variable states, shown here as i and z respectively. The variance values in the equation is represented as $\gamma(i)$, and therefore each value at i must be $\gamma(i) > 0$. The value $\beta(i)$ is a vector of real numbers which is the same size as the continuous variables in $PA(V)$ or the size of z, and these values are used to weight the distribution mean value.

4.4.11 CG Potential Operations

When performing computations on the clique tree using CG Potentials, there are a series of fundamental operations which can be utilised in order to perform all the necessary computations on the CG Potentials. The first operation is Extend and is critical in order for multiplication and division to occur. The extend operation is responsible for ensuring that the two CG potentials have a compatible and aligned scope to ensure that multiplication and division can occur.

4.4.12 CG Potential Extend

Given two CG potentials for which multiplication and division is to be performed, these two CG potentials must match in *scope*. The scope of a CG potential is defined by the variables which have been assigned to the CG potential.

Example 1. *Two canonical forms with their scope definitions and sample values*

$$\phi_1(A) = C\left(g_A, h_{\begin{pmatrix}A\end{pmatrix}}, K_{\begin{bmatrix}A\end{bmatrix}}\right)$$
$$\phi_1(A) = C\left(-5, (3), \begin{bmatrix}1\end{bmatrix}\right)$$

$$\phi_2(B, A) = C\left(g_B, h_{\begin{matrix}B\\A\end{matrix}}, K_{\begin{bmatrix}BB & BA\\AB & BB\end{bmatrix}}\right)$$
$$\phi_2(B, A) = C\left(-3, \begin{pmatrix}2\\-2\end{pmatrix}, \begin{bmatrix}1 & -1\\-1 & 1\end{bmatrix}\right)$$

In order to perform multiplication and division operations on these CG potentials, we first need to extend the scope of $\phi_1(A)$ by the scope of $\phi_2(A, B)$; the CG potential extend is done by simply adding zeros to the h vector and the K matrix given the order of the new extended scope:

$$\phi_1(A, B) = C\left(-5, \begin{pmatrix}3\\0\end{pmatrix}, \begin{bmatrix}1 & 0\\0 & 0\end{bmatrix}\right)$$

The two CG potentials ϕ_1 and ϕ_2 now have compatible variable scopes, and are ready for further operations.

4.4.13 CG Potential Initialisation

The first step in computing the clique tree, each clique must have CG potentials initialised for each assigned factor. Once the clique factors are initialised, the CG potentials for the factors must be extended and multiplied into a single CG potential which is associated to the clique. Initialising a CG potential is described in the below definitions:

$$g_A(i) = -\frac{\alpha(i)^2}{2\gamma(i)} - \log(2\pi\gamma(i))/2 \tag{4.1}$$

$$h_A(i) = \frac{\alpha(i)}{\gamma(i)} \begin{pmatrix} 1 \\ -\beta(i)' \end{pmatrix} \tag{4.2}$$

$$K_A(i) = \frac{1}{\gamma(i)} \begin{pmatrix} 1 & -\beta(i)' \\ -\beta(i) & \beta(i)\beta(i)' \end{pmatrix} \tag{4.3}$$

By way of an example, given a clique $C(A, B)$ which contains the two variable factors for A and B, we must first initialise each variable factor:

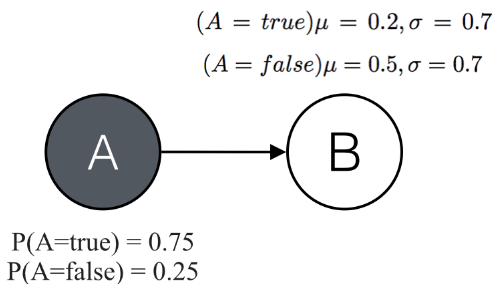


Figure 19: An example graph fragment with association conditional probability distributions

Example 2. *Initialising each variable in the example graph, will result in the below CG potentials:*

Factor	g	h	K
A.true	-1.46	0	0
A.false	-2.32	0	0
B(A.true)	0	0.29	1.43
B(A.false)	0	0.71	1.43

Table 5: Example potential initialised values for each variable factor

Then to create the CG potential for the clique $C(A, B)$, we need to multiply the factors for A and B. In order to achieve this, we must extend each variable factor CG potential *as detailed in 4.4.12. Once the multiplication has occurred the clique CG potential we be as shown in table 6.*

Factor	g	h	K
A.true, B	-1.46	0.29	1.43
A.false, B	-2.31	0.71	1.43

Table 6: Example clique CG potential initialised values

Definition 4.35. Vacuous Canonical Form

During the initialisation stage of the clique tree, all the separator CG Potentials are also initialised, however in this case the separators are initialised to zero or a *vacuous canonical form.* This canonical form is defined as $g = 0, h = 0, K = 0$

4.4.14 CG Potential Multiplication

Given two CG potentials that have the same scope, multiplication can be performed on them and is a simple operation of adding the two CG potentials together:
$\phi_1\phi_2 = (g_1, h_1, K_1) \times (g_2, h_2, K_2) = (g_1 + g_2, h_1 + h_2, K_1 + K_2)$

4.4.15 CG Potential Division

Given two CG potentials that have the same scope, division can be performed on them and is a simple operation of subtracting the two CG potentials:
$\phi_1/\phi_2 = (g_1, h_1, K_1)/(g_2, h_2, K_2) = (g_1 - g_2, h_1 - h_2, K_1 - K_2)$

Care must be taken when multiplying or dividing by what is terming a *vacuous canonical form*, this canonical form is analogous to the discrete factor initialised to all 1 values, and as such multiplication and division operations do not have any effect on the resulting canonical form.

If the situation arises whereby division by zero would occur, then the resulting CG potential should be a zero CG potential:

Given ϕ_1/ϕ_2, if $\phi_2 = 0$ then $\phi_1/\phi_2 = 0$

4.4.16 CG Potential Moment Form

When marginalizing on a CG potential in a hybrid network, in certain circumstances (see section 4.4.17) it is necessary to return the CG potential factor's values back to their normal or non-canonical form; this it known as the moment form. The moment form of a CG potential's canonical characteristics is denoted as $\{p, \xi, \sum\}$, this being in respect to the canonical characteristics $\{g, h, K\}$.

$$\sum(i) = K(i)^{-1} \tag{4.4}$$

$$\xi(i) = \sum(i)h(i) \tag{4.5}$$

$$p(i) \propto \left\{\det \sum(i)\right\}^{\frac{1}{2}} \exp\left\{g(i) + (h(i)' \sum(i)h(i))/2\right\} \tag{4.6}$$

4.4.17 CG Potential Marginalization

When marginalizing in a hybrid network with both discrete and continuous variables, marginalization must be performed initially over the continuous variables and then the discrete variables. When marginalizing over the discrete variables, we must first perform a check to establish if the variable factors being marginalized *to*, that is B within $A \setminus B$, are all discrete; or that the assigned factors in the CG potential that we are marginalizing *over* are independent of any discrete variables, i.e. the h and K part of the canonical characteristics are all equal. If this independence is true, then we can perform what is called *strong marginalization*, if the independence is false then *weak marginalization* must be performed instead.

When weak marginalization is to be performed, we must first convert the CG potential factors to their moment form (as described in section 4.4.16) prior to performing the marginalization. Once the marginalization is complete, the moment forms can be converted back to the CG potential form.

4.4.18 Message Passing - Absorption

As previously mentioned in section 4.4.9, within the message passing process of model calculation, the belief messages passed between cliques are absorbed into the receiving clique. The flow of the messages is established by the recursive message passing algorithm in both directions from the selected strong root clique; once a clique has initialised its CG potential and message passing begins, for each neighbouring clique the separator is constructed and a CG potential is constructed by marginalising from the sending clique's CG potential format to that of the separator. Figure 20 highlights the marginalization required to initialise the separator CG potential during both the collect and distribute phases of the message passing.

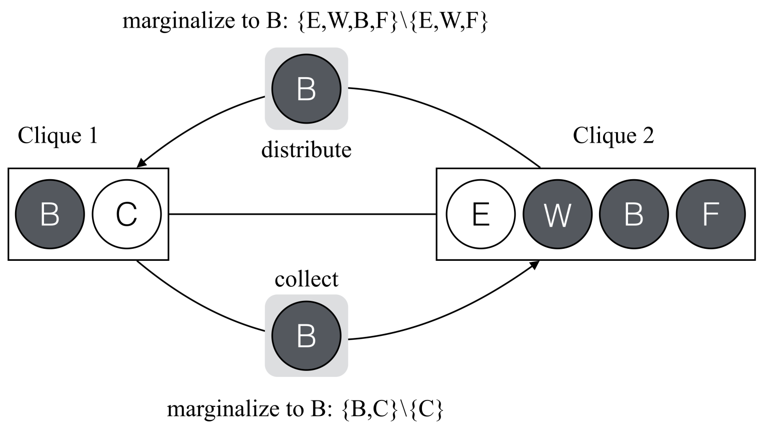


Figure 20: Separator CG potential marginalization example, with the strong root being clique 2.

Once the separator CG potential has been created after marginalization, the absorption of the separator CG potential into the receiving clique's CG potential is a matter of extending and multiplying the separator CG potential to that of the clique as discussed in sections 4.4.12 and 4.4.14

4.4.19 Evidencing

When specifying evidence for a variable in the Bayesian Network we are specifying either a specific discrete state is observed (that is with a probability of 100 percent); or that a continuous variable has been observed with a specified value. When performing discrete evidence in a hybrid Bayesian Network, the discrete variable state to be evidenced is removed from the CG potential factors in all cliques and separators which contain the evidenced variable. This removal of factors given evidence is described as *evidence reduction.*

When evidencing continuous variables, the evidence must also specify the observed numerical value of the evidence; like the discrete evidence the factor reduction in all CG potentials which contain the evidenced variable must be reduced. When evidence reduction is performed on for a continuous variable, such that $Y_\gamma = y^*_\gamma$, then the cardinality of the h vectors and K matrices will be reduced.

For example, if a CG potential has the canonical characteristics [8]:

$$h(i) = \binom{h_1(i)}{h_\gamma(i)}, K(i) = \begin{pmatrix} K_{11}(i) & K_{1\gamma}(i) \\ K_{\gamma 1}(i) & K_{\gamma\gamma}(i) \end{pmatrix}$$

then the reduced canonical characteristics of the CG potential will be [8]:

$$K^*(i) = K_{11}(i) \tag{4.7}$$

$$h^*(i) = h_1(i) - y^*_\gamma K_{\gamma 1}(i) \tag{4.8}$$

$$g^*(i) = g(i) + h_\gamma(i) y^*_\gamma - K_{\gamma\gamma}(i)(y^*_\gamma)^2/2 \tag{4.9}$$

4.5 Causal Bayesian Networks

A Causal Bayesian Network is a special class of Bayesian Networks, one which attempts to model the *nature* of the subject under the scrutiny of the modelling task. In order to model the nature, it is necessary to describe the variables as being either exogenous or endogenous; that is, influencing factors that are external to your control and factors which you are able to have some influence over. It is typically a convention to put any exogenous variables in a Causal Bayesian Network as parents to endogenous variables, as a rule it is the nature of exogenous variables that it is not possible to exert any influence on them by any endogenous variables in the model; indeed, it is quite often the case that exogenous variables are unobservable.

The influence that a set of variables have on another set of variables in the model is described by a set of probabilistic functions (see section 4.1). It is in these functions of the relationships between causal variables that set Causal Bayesian Networks apart from Bayesian Networks, in that they describe the nature by which the variables impart a causal influence from one to the other; these causal influence functions are representations of phrases that we use regularly to describe the world around us. For example, "X will be more likely if Y is true", "X causes Y" or "Y happened because of X", these utterances can be encoded in the conditional probability tables of the relationships between variables. Additionally, Causal Bayesian Networks allow the analysis of *counterfactuals*, that is answering statements such as "would X have occurred if Y had occurred differently".

5 Conditional Probability Elicitation

When designing Causal Bayesian Networks, it often the case that subject matter, or domain experts, are called upon to assist or own the design of the model. Given the focus of this thesis is on gaining insight into business domains and decision making, it is very likely that the domain expert would not be well versed in the process of eliciting probabilities to describe how parts of their business domain influence one another. Nevertheless, if exact inference is required to gain further insights, it is a necessary process but one which is increasingly susceptible to misjudgements in assessments and biases [10]. There are however, various methods by which given sufficient data these probabilities can be extracted via algorithms; these methods are designed to overcome human biases, however they can be quite time consuming to implement, particularly when dealing with large causal models. Furthermore, algorithm based extraction of probabilities do rely on having sound unbiased data, a concept which can be difficult to consistently find within businesses. Although with the advent of *big data*, more and more businesses take increasing interest in the collection and analysis of data. However, it is often the case that in order to amalgamate these disparate sources of data into a cohesive *data warehouse* takes significant effort, if it is at all possible. Therefore, when designing Causal Bayesian Networks, these issues must to taken into account when deciding the method by which probabilities are elicited.

During research it is apparent that the problem of generating probabilities from subject matter and domain experts is recognised, however it seems that the task of manually specifying these probabilities is less in focus. When the designer of a causal model is faced with the problem of specifying these probabilities, it is important that they must be aware of these potential biases, and also how to perform the elicitation of probabilities from domain experts. It is this area that this section gives focus, when a domain expert is asked by a designer or perhaps is designing the model themselves, is there a set of words which adequately describe the probabilistic degree of influence. This investigation is limited only to discrete probabilities, which are prone to the most variation and bias.

5.1 Heuristics and biases

When describing an assessment as being biased, it means that extra factors that are often irrelevant are included in the assessment, or that more relevant factors are somehow ignored, and these misaligned factors somehow skew the assessment in an undesirable way. When making an assessment about an event or response which is uncertain, particularly when using Causal Bayesian Networks, it is necessary to attribute continuous probability assessments to a discrete range of states which fully describe the event. Given this necessary assessment, how does a domain expert make these assessments in numerical form? It is via the subconscious use of heuristics which they use to effectively breakdown and simplify the task of assessment into simple judgemental operations. There are three main heuristics of bias which are commonly used: representativeness, availability and anchoring [10].

The *representativeness* heuristic describes the process whereby people make assessments which are evaluated by the degree to which one event is representative of another. The following example from a study by Tversky and Kahneman [10] demonstrates this heuristic:

"Steve is a very shy and withdrawn, invariably helpful, but with little interest in people, or in the world of reality. A meek and tidy soul, he has a need for order and structure, and a passion for detail." Which of the following professions would Steve be most likely employed in: farmer, salesman, airline pilot, librarian or physician?

As a result of this description, most people would choose the librarian option as the most likely job that Steve has; this kind of assessment can give rise to serious biasing as the similarity or representativeness of two concepts should not affect judgements of probability. The representativeness category of heuristic also has further problems that the example given, these are known as: *base rate insensitivity* and *sample size insensitivity.*

Base rate or prior probability insensitivity is when the underlying fact of a statistic is ignored in favour of representativeness; however, the prior probability has no direct effect on the assessor's judgements of stereotypes, in the above example it does not alter the stereotype that Steve is most likely to be a librarian.

Sample size insensitivity occurs when the assessor estimates the probability likelihood by the similarity of the estimate to the sampled value, essentially the estimate is independent of the sample size. The following example from a study by Tversky and Kahneman [10] demonstrates this heuristic:

"Imagine an urn filled with balls, of which 2/3 are of one colour and 1/3 of another. One individual has drawn 5 balls from the urn, and found that 4 were red and 1 was white. Another individual has drawn 20 balls and found that 12 were red and 8 were white. Which of the two individuals should feel more confident that the urn contains 2/3 red balls, and 1/3 white balls, rather that the opposite?"

The study resulted in more people feeling that the first individual was the stronger in evidence of the two, despite the odds for the first individual being 8 to 1 rather than 16 to 1 for the second individual.

The *availability* heuristic describes the process whereby assessors can bias the estimate of probabilities due to the ease by which instances or occurrences can be brought to mind, for example the probability that a new business process would not be very successful due to the difficulty in implementing it based on similar projects that had been implemented previously. A second factor to availability which can introduce biases is that more recent occurrences are much more likely be available as estimation evidence than earlier occurrences, irrespective of relevance. As an adjunct of this premise, familiarity of the subject being assessed can also be a factor. If a business project is being modelled and specific uncertainty assessment being made, then the assessor could easily be biased in their assessments if the project is one of their own, which they have a vested interest in.

The final heuristic of *anchoring* describes the process whereby assessor makes an uncertainty estimate starting from an initial value, more often than not based on representativeness and/or availability biases, and then adjusts this starting point to yield the final estimate. In most cases the adjustments made are insufficient and overall different initial values yield different estimates, which are fundamentally biased towards the initial value. The following example, which is based on an existing study by Tversky and Kahneman [10], demonstrates this heuristic:

Two groups of individuals from a business estimated the result of a numerical expression within 5 seconds, each group had no prior knowledge of the expression and the results were collected individually from each group. The first group were asked to evaluate the expression $8 \times 7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1$, and the second group were asked to evaluate the expression $1 \times 2 \times 3 \times 4 \times 5 \times 6 \times 7 \times 8$. In all cases the first group evaluated the result to be a value significantly higher than the second group.

5.2 Domain Experts

When it is necessary to use domain expert knowledge to assess the probabilities of a causal model, ideally it will be an expert who has the necessary domain knowledge and experience in or a familiarity with assessing uncertainty. Unfortunately, due to the very nature of these two disciplines, it is rarely the case that a single expert is available, rather the task of assessing uncertainty falls upon a pool of experts; ideally this pool of experts is as small as possible to mitigate the conflicts in assessing the probabilities. It is recommended that at least one of these experts would have been involved in the identification and construction of the model, so that any ambiguous variables can be explained sufficiently. When the assessment of the probabilities is being undertaken by a group of domain experts, research suggests that the optimal group size is around three members [11].

When probabilities are being assessed by domain experts the elicitation of precise probability numbers is far less desirable than using words to describe the degree of influence. This is attributed to the idea that single or very few descriptive words to express the degree of uncertainty of influence feels more natural in its encapsulation of the uncertainty. However, it is still very much the case that using words to describe uncertainty will have a varying degree of scope for different domain experts. Therefore, it is desirable to have a set of descriptive words, which aim to elevate up the scale by which a domain expert needs to estimate the influence probabilities; also this set of words provides a set standard by which the assessors can agree on and use to make probability estimates, which are much more aligned to a natural language method of assessment.

5.3 Words Of Probability

When a domain expert is called upon to either fully design or provide probability estimates for a Causal Bayesian Network, then the number of required estimates could potentially be quite overwhelming; if the model is primarily composed of discrete variables, then the volumes of probability distributions required can be large, continuous variables mitigate this somewhat given they are based on a single estimate per parent variable state space. Of course this can still pose an issue if the continuous variable in question has numerous discrete parent variables.

Therefore, it is desirable to have in place a method of quickly providing probability estimates to the model. Once the probabilities have been provided for the model, various sensitivity and validation analysis methods can be performed to further refine the probabilities. It is desirable that a set of verbal expressions is usable by a domain expert to specify these initial estimates; although it is equally desirable that the numeric assessment should still be made available for situations where accuracy is required. When people are required to provide estimates of probability they would much prefer to use verbal expressions when they have a lack of confidence in the specific area; the verbal expression feel much more natural, and allow them to express the uncertainty of their assessment.

In a business scenario various case studies have shown that the majority of causal models defined have consisted of primarily continuous variables; this could be attributed to the inherent financial and volume based modelling tasks in the business sector. Also, the constraint on continuous variables of only having continuous child variables means that once continuous variables are added to the model, in most cases it then becomes continuous from then on. During research the use of discrete variables has been mainly focused on describing an ordered ranked status of a specific concept, usually with the range Red, Amber and Green; other common usage of discrete variables is for binary statement of True of False. When defining discrete variables, it is common that the domain expert rarely defines more than five discrete states. Feedback has been shown that more than five states results in the causal model being described as "limited in movement" around the discrete variables in question; the domain experts also expressed a positive correlation of difficulties in expressing the probability estimates given a higher number of discrete variable states.

Extensive research has been performed [12] on possible ranges of verbal expressions for probability analysis; however for the purposes of this research it was decided that the range of words should be limited to a range of seven words, this decision was arrived at partly from experience of creating causal models for numerous test cases; also it corresponded with research performed around the concept that the optimal number of bits of information a human being can adequately cope with is seven plus/minus two [13]. It also was decided that too large a set would result in negating the purpose of the words, i.e., providing words designed to span every ten percent of the probability range would result in the domain experts having difficulty resolving one word from another. Having seven words means that we can have the two extremes of the range, zero and one hundred percent and the mid-point, and then have two additional words between the mid-point and extremities. Table 7 shows the range of the probability estimate words.

Probability words
impossible
very unlikely
fairly unlikely
fifty-fifty
fairly likely
very likely
almost certain

Table 7: The selected verbal expressions of probability

5.3.1 Words of probability survey

A survey to the students of York university was carried out in order to establish quantitative values given various scenarios to describe the words of probability as listed in Table 7. The specific scenarios are listed in Appendix B, each question describes a specific scenario, and the respondent must position a numeric slider for a range between 0 and 100 to where they feel the value best describes the condition of the scenario. Additional to these scenarios questions, the respondents were asked to order the words of probability in ascending order from least likely to most likely.

129 people responded to the survey request, the respondents were composed of Computer Science students and a mix of students from other departments from the University of York, out of which 19 respondents only partially completed the survey. The average time taken to complete the survey was 1047 seconds.

Analysis of the ordering survey question shows that out of 129 respondents that took part in the survey to some degree, a total of 115 respondents completed this question. Grouping the results into the various orderings, there are 10 distinct groupings of different word orderings. 86 respondents correctly ordered the words in ascending order and 19 respondents ordered the words correctly albeit in the wrong direction, this is assumed to be a mis-reading of the ordering instruction as specified in the question. The remaining 10 respondents gave an ordering of the words to approximately 70 percent correctness with a mix of ascending and descending orders.

The scenario question "*Matt places 6 RED balls into an empty bag, he then passes the bag to you and asks you to remove a ball from the bag. It is IMPOSSIBLE that you will remove a BLUE ball from the bag*" which is designed to elicit a response to the Impossible scenario yielded mixed results. Analysing the response data, the results are 37 responses with the value 100, 23 responses with the value 0, 1 response of 99, 50 and 2. This question was designed to elicit a response close to 0, it was decided that the 37 responses with the value of 100 are due to a different interpretation of the question, i.e., the interpretation was centred around the chance of the scenario being 100 percent, in this case 100 percent chance of it being impossible to retrieve a blue ball.

This pattern of interpretation was not present in any of the other questions except for the Fifty-Fifty question, whereby out of a total of 108 responses, 19 of which responded with a value of 100 percent. Cross referencing the Impossible and Fifty-Fifty responses, there are 63 respondents who completed both questions of which 17 exhibited the potential interpretation problem.

Interestingly the Almost Certain scenario was interpreted as expected with a response close to 100 percent, therefore this behaviour could be explained by the responses being aligned to the truth of the scenario statement, rather than the questioned expression of chance given the scenario. It is also interesting that this pattern of interpretation is only present in the 3 scenarios which are more aligned with a less uncertain response, i.e., Almost Certain, Fifty-Fifty and Impossible. Table 8 shows the average aggregated and standard deviation results of the survey responses.

Probability word values		
Word	Value	Standard Deviation
impossible	0.0003	0.0025
very unlikely	0.2879	0.3000
fairly unlikely	0.3593	0.1776
fifty-fifty	0.51	0.0552
fairly likely	0.6726	0.1430
very likely	0.7779	0.1797
almost certain	0.9451	0.0774

Table 8: The surveyed average probability values for the verbal expressions

5.4 Architecture

The design for a system which utilises Causal Bayesian Networks within a business environment is built on the principle that a business can describe the major entities or concepts which constitute their business and its processes; this element of the system is called the business data landscape, in that by itself it is a point of reference for all captured business concepts, whether there is data for these concepts or not. Each entity in the business data landscape is described by a set of properties, similar to the entities these properties can have recorded data assigned to them or they can be modelled via a Causal Bayesian Network.

A system user can interface directly with the business data landscape by connecting multiple entities together; when entities are connected via an association relationship any data in the entities is filtered by intersection by default. Conditional predicates can be applied to connected entity properties in order to further filter the data being shown in all connected entities. This adhoc capability to connect and filter the data landscape is a very powerful method for a system user to gain an initial insight into the observed data in the landscape.

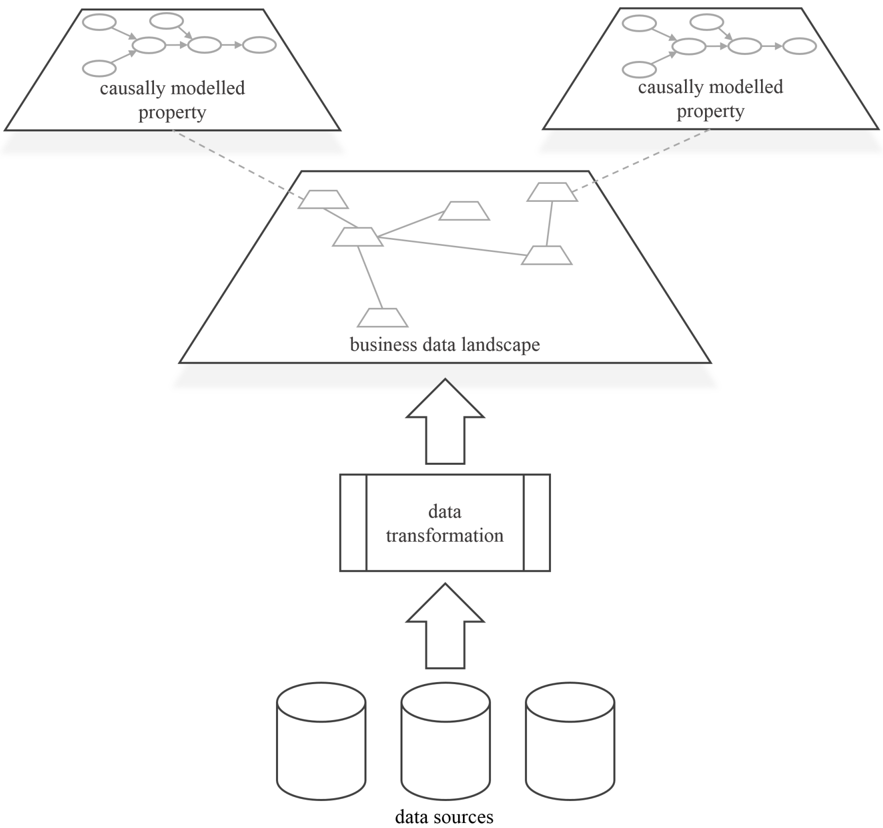


Figure 21: Overall system architecture.

5.4.1 Landscape Elements

Each entity contained within the business data landscape is a distinct representation of a business concept, for example a Department or a Project; these entities encapsulate the business concept and all properties which describe it. The properties on an entity can be of two types: a data property and a causally driven property. A data property should exist for every observable attribute of a business concept; in this way each business entity becomes measurable.

Entities within the landscape can have data associated with them, as shown in figure 21. Data for disparate external data sources can be extracted and transformed into the appropriate business data landscape entities, however, it is also valid for landscape entities to contain no data.

5.4.2 Causally Driven Properties

A causally driven property can exist on an entity, and these properties represent the output of a specific variable in an associated Causal Bayesian Network. These properties enable a business analyst to elaborate on the drivers of a property of a business concept, and as such gain a deeper understanding into potential influencing factors which can be changed. It is the intention that the causally driven properties can also be created for existing data properties, this gives the ability to causally model a business concept's observed property, again for potential change efficiencies.

The associated Causal Bayesian Network for a causally driven property can have key model variables be data bound to a specific landscape entity data property. This enables dynamic recalculation of the causally driven properties, when adhoc querying and filtering is applied to the business data landscape. Figure 22 highlights the flow of observational data from key business data landscape entity properties through to data bound causal variables; once the causal model is calculated with these data bound observations, then a specific causal model variable is providing the calculated value back to a business data landscape entity property.

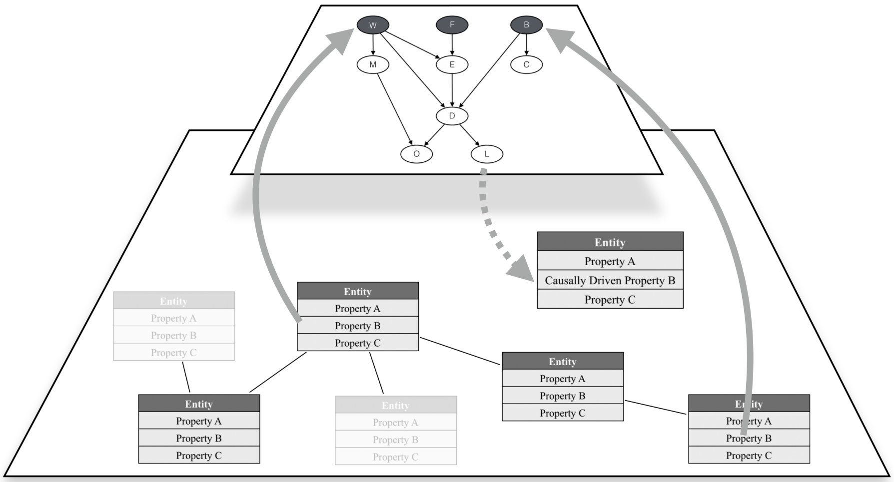


Figure 22: Business data landscape data bound causal model variables, and a causally driven property. The faded business data landscape entities are not selected.

5.5 Conditional Probability Value Generation

When a domain expert is tasked with creating a Causal Bayesian Network, they have the task of specifying the probability distributions to describe the nature of the causal relationships, once the network of variables has been mapped out. Also mentioned are the problems around introducing biases when specifying probability numbers into these distributions. It is more desirable if the domain experts could specify, in a qualitative manner, the overall nature of the causal relationships between the variables and then have a system generate the probability distributions. The idea behind a guided probability distribution is to have a system generate an initial hypothesis about the causal relationships in the network given the relatively little information a domain expert may have about them. Given that Causal Bayesian Networks, which are supporting strategic decision making, can become large and complex, outside of data analysis the question of how the probability distribution acquisition can be performed in these complex domains is the problem which needs to be addressed.

The specifics of generating probability distributions is explored in this section, specifically limited to the generation of probability distributions for discrete variables. The focus on discrete variables is borne out of the nature of the distributions; the decision to focus solely on discrete variables is due to the fact that given a complex network of discrete variables with even small event spaces, the volume of required probability numbers can be very large. When these distribution spaces are of a large size, it can be extremely difficult to manually specify probabilities, even for someone who knows in great detail how the causal relationship should behave. Therefore the domain expert should only be required to indicate how pairs of discrete variables interact and to what degree in order to generate the probability distribution.

The proposed method of generating these discrete distribution is with the utilisation of constraint generation, and then solving the constraint graph to produce the probabilities. This method differs from typical commercially available Bayesian Network systems, for example Hugin, in that these systems either allow the user to manually enter the distributions or have them computed via simple expressions, which still feel like an unnatural method for a domain expert. The method of constraint generation is reliant on the discrete variables event space being in a ranked order, and with mapping information from parent variable event spaces to the child variables event space, simple constraints can be generated.

In the following section I explore the concepts of Constraint Satisfaction Programming, and introduce the concepts of constraints and a method of solving constraint problems and how this method is utilised to generate a probability distribution for a discrete variable.

5.6 Constraint Satisfaction Problems

A Constraint Satisfaction Problem (CSP) is simply a mathematical problem whereby given a series of defined constraints, a general purpose constraint solver can find a solution which assigns a value to one or more variables which satisfy all the defined constraints. CSPs are widely used in systems that deal scheduling, planning and even logic puzzles such as Sudoku or crosswords. CSPs are comprised of a series of variables and constraints upon them, a constraint solver aims to find an assignment of values to the variables so that the constraints are satisfied. These constraint solvers utilise various algorithms to *search* the space of possibilities that the variable assignments could be, the set of values that could be assigned to each variable is known as the *domain* of the variable. For example, when finding a solution to a game of Sudoku, the domain of each cell would be a set of integer numbers in the range 1 to 9, and the constraint on each column and row of the game grid is that each cell must be different in the column and row. In this example, given a Sudoku grid which has been either partially completed or is blank, it is possible for a constraint solver to systematically search the solution space, assign values from domains to each grid cell and ensure that the constraints are satisfied, if a solution is possible then a solution will be found.

In the case of a discrete variable in the Causal Bayesian Network, the challenge of generating the probabilities is a similar constraint satisfaction problem to the Sudoku example. Table 9 shows an example of a typical condition probability distribution for a discrete variable which has two discrete parents, all variables have an event space of red, amber and green.

		red	amber	green
red	red	0.95	0.026	0.024
red	amber	0.713	0.186	0.101
red	green	0.534	0.302	0.164
amber	red	0.05	0.713	0.237
amber	amber	0.007	0.95	0.043
amber	green	0.08	0.534	0.386
green	red	0.143	0.323	0.534
green	amber	0.089	0.199	0.712
green	green	0.018	0.032	0.95

Table 9: Example discrete conditional probability distribution

A constraint on each row of a conditional probability distribution is that each row must sum to exactly 1, the example shown in table 9 is for two parents which influence positively on the child variable. Therefore it is also possible to assign constraints to each cell which provide rules for how each row cell is constrained given values assigned to its row peer cells. This idea forms the basis of generating the probability distributions for a domain expert.

Definition 5.1. Constraint Satisfaction Problem

A CSP is a triple $P = \langle X, D, C \rangle$ where X is an n-tuple of variables $X = \langle x_1, x_2, ..., x_n \rangle$, D is a corresponding n-tuple of domains $D = \langle D_1, D_2, ..., D_n \rangle$ such that $x_i \in D_i$, C is a t-tuple of constraints $C = \langle C_1, C_2, ..., C_t \rangle$.

A constraint C_j is a pair $\langle R_{S_j}, S_j \rangle$ where R_{S_j} is a relation on the variables in $S_j = scope(C_i)$. R_i is a subset of the Cartesian product of the domains of the variables in S_i. [14]

5.7 Constraints

For the generation of conditional probability distributions, it is assumed that the variable domains are mapped on the finite set $\mathbb{Z}$ of integers which are discretized within the range from 0.001 to 0.999.

Definition 5.2. Constraint

A constraint c is a relation defined on a sequence of variables $X(c) = (x_{i_1}, ..., x_{i_{|X(c)|}})$, called the *scheme* of c. c is the subset of $\mathbb{Z}^{|X(c)|}$ that contains the combinations of values (or tuples) $\tau \in \mathbb{Z}^{|X(c)|}$ that satisfy c. $\mathbb{Z}^{|X(c)|}$ is called the *arity* of c. Testing whether a tuple τ satisfies a constraint c is called a *constraint check*. [14]

Definition 5.3. Constraint network

A constraint network is comprised of:

- a finite sequence of variables $X = (x_1, ..., x_n)$,
- a domain for X, that is, a set $D = D(x_1) \times ... \times D(x_n)$, where $D(x_i) \subset \mathbb{Z}$ is the finite set of values that variable x_i can take, and
- a set of constraints $C = c_1, ..., c_e$, where variables in $X(c_j)$ are in X.

[14]

A constraint network can be associated with a graph, where the variables are nodes and the schemes of constraints are edges.

5.8 Constraint Generation

In order to generate constraints, a set of configuration data must be setup for each discrete variable for which probability generation is to be performed. Given that the designer of the Bayesian Network could be a domain expert, the aim is to keep the configuration simple and minimal enough to generate a conditional probability distribution that is a good starting point or hypothesis. For each discrete variable in the model the domain expert must assess the question "*Given the parental influence, what should be the effect on the child states?*". In order to generate a probability distribution that addresses this question, the domain expert will initially be required to indicate a parent variable ordering for the importance of influence over the child variable. This ordering will allow the domain expert to specify which parent should have the largest influence on the child, which is ordering rank 1, down to the least influential at rank n. The parent variable ordering is defined as the *influence rank*.

Each discrete variable that requires probability generation must have its discrete states numerically ranked in ascending order of best to worst. For example, given a discrete variable with the states Green, Amber and Red the rankings for these states could be 1, 2 and 3 respectively. The *state ranks* enable constraint generation to effectively decide which states should be greater or less than other states in probability value.

The next configuration data is a *state mapping* between parent variable state tuples and specific child variable states, each individual state map defines an *anchor* within the conditional probability distribution table, each *anchor* is converted to a unary constraint during constraint generation. The *anchor* configuration is a representation of the *anchoring* heuristic as detailed in section 5.1, when constraint generation is performed the constraints are started from the *anchor* cell in the CPD.

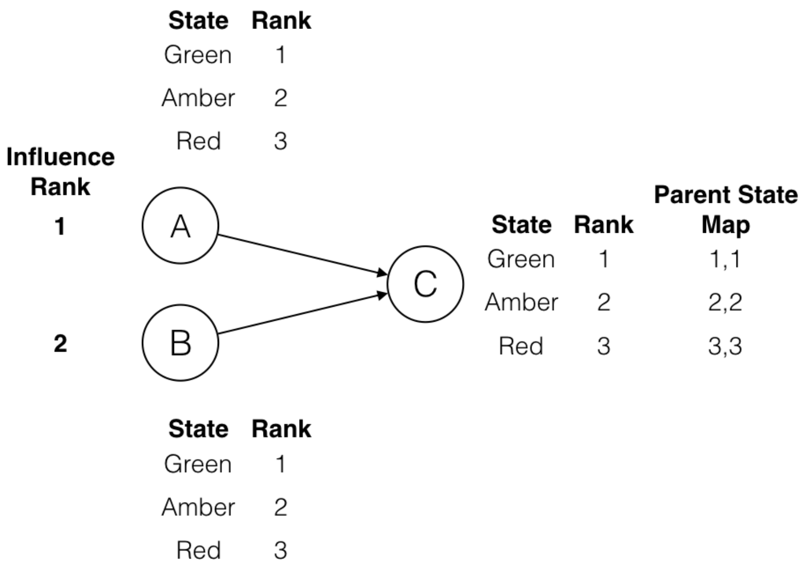


Figure 23: Example constraint generation ranking configuration.

Fig. 23 shows an example Bayesian Network fragment along with its constraint generation configuration. Each variable A, B, C has the same ranked state set $\{red, amber, green\}$. The parent variables $Pa(C) = A, B$ have the influence ranks $A = 1$ and $B = 2$ to indicate that variable A is likely to impart more influence on C than B does. The state maps for variable C are also shown, in this example three state maps are defined for each state on variable C.

The final configuration is to set the degree of influence for the *anchors* by choosing one of the words of probability as detailed in Part 2. The range of words available is dependent on the number of discrete states on the child variable, given $P = \{AlmostCertain = 0.9451, VeryLikely = 0.7779, FairlyLikely = 0.6726, Fifty - Fifty = 0.51, Fairly - Unlikely = 0.3593, VeryUnlikely = 0.2879, Impossible = 0.0003\}$ then $W = P > \frac{1}{|s|}$ where W is the set of available words of probabilities.

It is important to note that the generated conditional probabilities using this method are all valid given that a constraint problem solution will be found. However, the case of whether the resulting probabilities match the expectations of the domain expert is something that requires some analysis of final model as a whole. It is also assumed that the Causal Bayesian Network structure is correct and fixed. The complexity of both generating the constraints and then solving the constraint problem can be computationally demanding given the number of parent variables, and the number of discrete states on each parent. The CPD distribution and generated constraints can exponentially increase to hundreds of parameter values, and so this must be considered during the design of the causal model.

5.8.1 Conditional Probability Distribution Table - Constraint Framework

In order to aid effective constraint generation given the aforementioned configurations, the conditional probability distribution table will be partitioned into *influence blocks*. Table 10 shows a typical representation of the basic structure of a discrete variable conditional probability table. The number of rows in the table is determined by the Cartesian product of the number of variable parents and the discrete states on these parents. The columns for the child CPD variable in question are determined by its discrete states.

		Child Variable	
		State 1	State 2
Parent 1 States	Parent 2 States	Probability	Probability
...	...	...	...

Table 10: Example CPD fragment

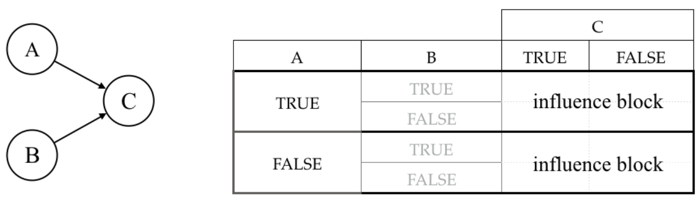


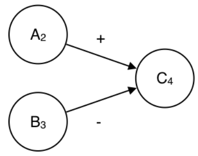
Figure 24: Example influence blocks

An *influence block* within the CPD table is defined by the configuration of the most influential parent variable, that is the parent variable with influence rank 1. Within the CPD table the order from left to right of the parent variables is determined by the influence rank configuration of the parent variables. Fig. 24 highlights an example of influence blocks given a simple model fragment for the variable C with parent variables A and B. All variables have binary states,and parent variable A has the influence rank of 1.

Fig. 25 highlights various network variable fragments and the CPD tables for each given different influence ranking of parent variables. Fig. 25a and 25b show the same network fragment with the influence ranks of $A = 1, B = 2$ for Fig. 25a and $B = 1, A = 2$ for Fig. 25b. The constraint notation used in Fig. 25 for each CPD table cell is in the format of a CPD cell coordinate. For example Fig. 25a in the top-right most CPD cell has the constraint <1,2,1 >1,2,2, which means this cell, with coordinate 1,1,1, should have a value less than the value in CPD cell with the coordinate 1,2,1 and a have a value greater than the value in CPD cell 1,2,2, and so on.

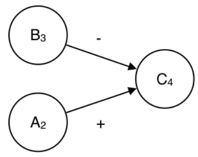
Each *influence block* must be assigned a single *anchor* cell reference, the cell coordinate for the *anchor* is specified by the *state mapping* configuration.

		C			
A	**B**	**1**	**2**	**3**	**4**
1	**1**	<1,2,1 >1,2,2	<1,1,1 >1,2,2	<1,1,2 >1,2,3	<1,1,3 >1,2,4
1	**2**	<1,3,1 >1,3,2	<1,2,1 >1,3,2	<1,2,2 >1,3,3	<1,2,3 >1,3,4
1	**3**	**A**	<1,3,1	<1,3,2	<1,3,3
2	**1**	<2,1,2	<2,1,3	<2,1,4	**A**
2	**2**	<2,2,2 >2,1,2	<2,2,3 >2,1,2	<2,2,4 >2,1,3	<2,1,4 >2,1,3
2	**3**	<2,3,2 >2,2,1	<2,3,3 >2,2,2	<2,3,4 >2,2,3	<2,2,4 >2,2,3



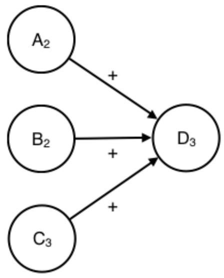
(a) Variable A more influential on C than B.

		C			
B	**A**	**1**	**2**	**3**	**4**
1	**1**	<1,1,2 >1,2,1	<1,1,3 >1,2,2	<1,1,4 >1,2,3	<1,2,4 >1,2,3
1	**2**	<1,2,2	<1,2,3	<1,2,4	**A**
2	**1**	>2,1,2	<2,2,2	<2,1,2 >2,2,2	<2,1,3 >2,2,4
2	**2**	>2,2,3	**A**	<2,2,2	<2,2,3
3	**1**	**A**	<3,1,1	<3,1,2	<3,1,3
3	**2**	<3,1,3 >3,1,2	<3,2,1 >3,1,2	<3,2,2 >3,1,3	<3,2,3 >3,1,4



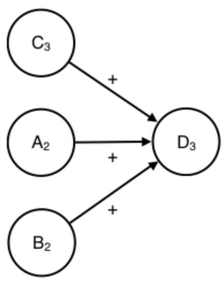
(b) Variable B more influential on C than A.

			D		
A	**B**	**C**	**1**	**2**	**3**
1	**1**	**1**	**A**	<1,1,1,1	<1,1,1,2
1	**1**	**2**	<1,1,1,1 >1,1,1,2	<1,1,2,1 >1,1,1,2	<1,1,2,2 >1,1,1,3
1	**1**	**3**	<1,1,2,1 >1,1,2,2	<1,1,3,1 >1,1,2,2	<1,1,3,2 >1,1,2,3
1	**2**	**1**	<1,1,3,1 >1,1,3,2	<1,2,1,1 >1,1,3,2	<1,2,1,2 >1,1,3,3
1	**2**	**2**	<1,2,1,1 >1,2,1,2	<1,2,2,1 >1,2,1,2	<1,2,2,2 >1,2,1,3
1	**2**	**3**	<1,2,2,1 >1,2,2,2	<1,2,3,1 >1,2,2,2	<1,2,3,2 >1,2,2,3
2	**1**	**1**	<2,1,1,2 >2,1,2,1	<2,1,1,3 >2,1,2,2	<2,1,2,3 >2,1,2,2
2	**1**	**2**	<2,1,2,2 >2,1,3,1	<2,1,2,3 >2,1,3,2	<2,1,3,3 >2,1,3,2
2	**1**	**3**	<2,1,3,2 >2,2,1,1	<2,1,3,3 >2,2,1,2	<2,2,1,3 >2,2,1,2
2	**2**	**1**	<2,2,1,2 >2,2,2,1	<2,2,1,3 >2,2,2,2	<2,2,2,3 >2,2,2,2
2	**2**	**2**	<2,2,2,2 >2,2,3,1	<2,2,2,3 >2,2,3,2	<2,2,3,3 >2,2,3,2
2	**2**	**3**	<2,2,3,2	<2,2,3,3	**A**



(c) Variable influence ordering of A, B, C for D.

			D		
C	**A**	**B**	**1**	**2**	**3**
1	**1**	**1**	**A**	<1,1,1,1	<1,1,1,2
1	**1**	**2**	<1,1,1,1 >1,1,1,2	<1,1,2,1 >1,1,1,2	<1,1,2,2 >1,1,1,3
1	**2**	**1**	<1,1,2,1 >1,1,2,2	<1,2,1,1 >1,1,2,2	<1,2,1,2 >1,1,2,3
1	**2**	**2**	<1,2,1,1 >1,2,1,2	<1,2,2,1 >1,2,1,2	<1,2,2,2 >1,2,1,3
2	**1**	**1**	>2,1,1,3	<2,1,2,2	<2,1,1,2 >2,1,2,3
2	**1**	**2**	>2,1,2,3	<2,2,1,2	<2,1,2,2 >2,2,1,3
2	**2**	**1**	>2,2,1,3	<2,2,2,2	<2,2,1,2 >2,2,2,3
2	**2**	**2**	>2,2,2,3	**A**	<2,2,2,2
3	**1**	**1**	<3,1,1,2 >3,1,2,3	<3,1,1,3 >3,1,2,2	<3,1,2,3 >3,1,2,2
3	**1**	**2**	<3,1,2,2 >3,2,1,3	<3,1,2,3 >3,2,1,2	<3,2,1,3 >3,2,1,2
3	**2**	**1**	<3,2,1,2 >3,2,2,3	<3,2,1,3 >3,2,2,2	<3,2,2,3 >3,2,2,2
3	**2**	**2**	<3,2,2,2	<3,2,2,3	**A**



(d) Variable influence ordering of C, A, B for D.

Figure 25: Example model fragments highlighting *CPD blocks*,*anchors* and *constraints*.

Definition 5.4. Constraint Generation Meta-Data

Let B be a Causal Bayesian Network with a directed acyclic graph structure $G(V, E)$ where the discrete variables V of the graph are Δ. Let $Z =\subset \Delta$ be the set of discrete variables which require probability generation. The constraint generation meta-data $A = (R, B, M, D)$ for each variable in Z is comprised of:

- an ascending ordered set of *influence rank* values $R = (r_1, ..., r_k)$ where $k = |Pa(Z_i)|$, one for each of the parents $Pa(Z_i)$ to indicate the order of parental influence for Z_i.

- Let I be an *influence block* and C is a set of individual probability values contained within each I. The $|C|$ of C is governed by the cartesian product of the number of discrete states of $Pa(Z_i) \setminus Pa(Z_{r_1})$, where $Pa(Z_{r_1})$ is the parent of Z with *influence rank* $= 1$.

- a set of I influence blocks $B = (b_1, ..., b_k)$ where $k =$ number of discrete states of the parent variable with influence rank r_1.

- a set of state mappings $M = (m_1, ..., m_k)$ where $k = |B|$, each state map routes a parent states tuple to a CPD influence block B_i to specify the anchor a_i. Each CPD influence block B_i has an associated anchor a_i.

- a set of degree of influence values $D = (d_1, ..., d_k)$ where $k = |B|$, selected from a restricted set of probability words which are applied to each corresponding a_i.

Algorithm 3 highlights the function to generate constraints for the cells in a CPD *influence block* which are neighbouring to an anchor cell. Figure 26 highlights an example of these *neighbour anchor cells*.

		C			
A	B	1	2	3	4
1	1	**Anchor**			
1	2	Neighbour Anchor			
1	3	Neighbour Anchor			
2	1			Neighbour Anchor	
2	2			**Anchor**	
2	3			Neighbour Anchor	
3	1			Neighbour Anchor	
3	2			Neighbour Anchor	
3	3			**Anchor**	

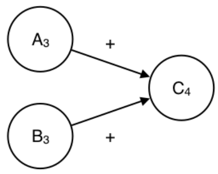


Figure 26: Anchor neighbouring cells

Algorithm 3: CPD constraint generation algorithm for anchor neighbour cells

Input: s:variable state; z comparison variable state

```
function GenerateConstraintsForAnchorNeighbour(s, z)
    if rank of s is min or max rank for the state then
        C ← set of constraints for s
        C1 ← new less than z constraint
        c ← parent state set of z
        r ← state rank of z
        n ← 0
        if r is the min state rank then
            n ← 1
        end
        if r is the max state rank then
            n ← -1
        end
        e ← new cell coordinate from c + n
        C2 ← new greater than e constraint
    end
    else
        C ← set of constraints for s
        C1 ← new less than z constraint
    end
```

Algorithm 4 highlights the function to generate constraints for each individual *cell* in a CPD, a cell equates to each state s_i of each discrete variable $Z_i \in Z$. Example results of this algorithm are shown in Fig. 25. The function *GenerateConstraintsForVariable* is initially called for each *anchor* cell in the CPD, then for each cell in the neighbouring rows within each anchors CPD block as shown in algorithm 4.

Algorithm 4: CPD variable constraint generation algorithm

Input: s:variable state; z comparison variable state; S: all variable states for the variable; p: previous CPD row anchor coordinate

function *GenerateConstraintsForVariable*(s, z, S, p)

$N \leftarrow S \setminus s$

if *rank of s is min or max rank for the state* **then**

foreach $x_i \in N$ **do**

$o \leftarrow$ CPD cell coordinate of s

$C \leftarrow$ set of constraints for x_i

$C_i \leftarrow$ new less than z constraint

if *x_i is not in the same CPD column as s and $p \neq o$* **then**

$C_i \leftarrow$ new greater than p constraint

end

$z \leftarrow x_i$

end

end

else

$H \leftarrow$ states in N with state rank higher then state rank of s, ordered by state rank ascending

$L \leftarrow$ states in N with state rank lower then state rank of s, ordered by state rank descending

foreach $h_i \in H$ **do**

$C \leftarrow$ set of constraints for h_i

$C_i \leftarrow$ new less than z constraint

if *h_i is not in the same CPD column as s and h_i is not in the same CPD row as s* **then**

$C_i \leftarrow$ new greater than p constraint

end

$z \leftarrow h_i$

end

$C \leftarrow$ set of constraints for L_1

$C_i \leftarrow$ new greater than H_1 constraint

$z \leftarrow L_i$

$T \leftarrow L \setminus L_1$

foreach $t_i \in T$ **do**

$C \leftarrow$ set of constraints for t_1

$C_i \leftarrow$ new less than z constraint

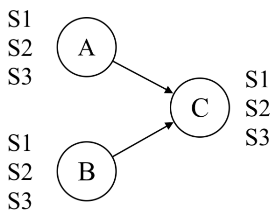
$z \leftarrow t_i$

end

end

5.8.2 Example constraint generation algorithms

Below are examples of algorithms 3 and 4 given a simple variable structure as shown in Figure 27, all variables have three discrete states $\{1, 2, 3\}$. The example algorithm execution is for generating the constraints for the first two rows of the *influence block* as shown in Figure 27, in order of CPD rows starting from the top at $A.1, B.1$ and then $A.1, B.2$.



		C		
A	B	C.1	C.2	C.3
A.1	B.1	**A**		
	B.2			
	B.3			
...	...	...	...	...

Figure 27: Example CPD fragment to illustrate an example of algorithms 3 and 4

Algorithm 4 Example		
Variable	Value	Stage
s	$[\{A.1, B.1\}, C.1]$	initialisation
S	$[\{A.1, B.1, C.1], [\{A.1, B.1\}.C.2], [\{A.1, B.1\}, C.3]\}$	initialisation
N	$\{[\{A.1, B.1\}.C.2], [\{A.1, B.1\}, C.3]\}$	initialisation
z	$[\{A.1, B.1\}, C.1]$	initialisation
p	$[\{A.1, B.1\}, C.1]$	initialisation
x_1	$[\{A.1, B.1\}, C.2]$	iteration 1
o	$[\{A.1, B.1\}, C.1]$	iteration 1
C	$\{[[\{A.1, B.1\}, C.2] < [\{A.1, B.1\}, C.1]]\}$	iteration 1
z	$[\{A.1, B.1\}, C.2]$	iteration 1
x_2	$[\{A.1, B.1\}, C.3]$	iteration 2
o	$[\{A.1, B.1\}, C.1]$	iteration 2
C	$\{[[\{A.1, B.1\}, C.3] < [\{A.1, B.1\}, C.2]]\}$	iteration 2
z	$[\{A.1, B.1\}, C.3]$	iteration 2

Table 11: CPD cells for influence block cells $[\{A.1, B.1\}, C.2]$ and $[\{A.1, B.1\}, C.3]$.

Algorithm 3 Example		
Variable	Value	Stage
s	$[\{A.1, B.2\}, C.1]$	initialisation
z	$[\{A.1, B.1\}, C.1]$	initialisation
C_1	$\{[[\{A.1, B.2\}, C.1] < [\{A.1, B.1\}, C.1]]\}$	
c	$[\{A.1, B.1\}]$	
r	1	
n	1	
e	$[\{A.1, B.1\}, C.2]$	
C_2	$\{[[\{A.1, B.2\}, C.1] > [\{A.1, B.1\}, C.2]]\}$	

Table 12: CPD constraint for influence block anchor neighbour cell $[\{A.1, B.2\}, C.1]$.

Algorithm 4 Example		
Variable	Value	Stage
s	$[\{A.1, B.2\}, C.1]$	initialisation
S	$\{[\{A.1, B.2\}, C.1], [\{A.1, B.2\}.C.2], [\{A.1, B.2\}, C.3]\}$	initialisation
N	$\{[\{A.1, B.2\}.C.2], [\{A.1, B.2\}, C.3]\}$	initialisation
z	$[\{A.1, B.2\}, C.1]$	initialisation
p	$[\{A.1, B.1\}, C.1]$	initialisation
x_1	$[\{A.1, B.2\}, C.2]$	iteration 1
o	$[\{A.1, B.2\}, C.1]$	iteration 1
C	$\{[[\{A.1, B.2\}, C.2] < [\{A.1, B.2\}, C.1]], [[\{A.1, B.2\}, C.2] > [\{A.1, B.1\}, C.2]]\}$	iteration 1
z	$[\{A.1, B.2\}, C.2]$	iteration 1
x_2	$[\{A.1, B.1\}, C.3]$	iteration 2
o	$[\{A.1, B.1\}, C.1]$	iteration 2
C	$\{[[\{A.1, B.1\}, C.3] < [\{A.1, B.1\}, C.2]]\}$	iteration 2
z	$[\{A.1, B.1\}, C.3]$	iteration 2

Table 13: CPD cells for influence block cells $[\{A.1, B.2\}, C.2]$ and $[\{A.1, B.2\}, C.3]$.

Part II

6 Case Study

The case study for the application of the techniques described in this thesis, was selected based upon a common pattern of requirements by business; that is the analysis and adaptation of provided capacity to meet a demand upon the business. This demand / capacity pattern is applicable to a wide range of business requirements, and can equally be applied to people capacity or manufacturing capacity management. The chosen case study is a representative instance of this capacity demand problem as applied to a fictitious bank, which I have named the *Bank of the States*, any similarities within the specifics of the data to any real world banking organisations is purely coincidental. The main purpose of this case study is to demonstrate the application of the artificial intelligence techniques described in this thesis; how they can provide a deeper understanding of the key influencing factors to the specific use case and then to model options for change within the business.

The specific area to be used within the Bank of the States is ultimately focused on the bank selling mortgages to the general public, and those mortgage applications being successful through to completion, and so providing revenue for the bank. The causal model in the case study is designed to describe the key social and economic factors which influence the general public to approach the bank for a mortgage. I have based the Bank of the States in the United States of America, and so the social and economic factors data has been gathered and compiled from US Census data for each State and County.

The causal model describes a scenario which the Bank of the States is currently in, whereby they are seeing poor satisfaction from both their customers and their staff with regard to their internal and external systems. The level of satisfaction is further complicated due to the age of customers wanting different needs from the banking systems, the younger population is very keen to increase the availability of an online presence, and would rather perform all activities online without the need to visit a local branch. Although the older population prefers the more human face to face aspect of banking. The Bank of the States has also been witnessing a slow reduction of new customer mortgage applications over the past 18 months, and suspects that the overall customer satisfaction and market presence requires some attention. Various competitors of the bank have been increasing their market share and online banking presence, and whilst at this stage the bank is still achieving new customers, the rate of decline means the bank must address these satisfaction and confidence issues sooner rather than later. The bank has decided that it should run projects to ultimately address the customer satisfaction issue, however it is uncertain to them, given options of an online systems upgrade and a fresh marketing initiative, in what specific areas these projects help and what degree. Also, if the projects are successful, what demands of staffing levels will these changes have?

The intended purpose of this case study is to help the Bank of the States to understand the effects of performing a mortgage application and monitoring system upgrade and/or initiating a fresh marketing campaign for the bank, and what levels of capacity the bank should expect given these potential changes.

The design of the causal model for the Bank of the States' case study was created by myself acting as the domain expert for the fictitious bank, however the influencing factors and the general degree of influence is based upon realistic economic factors and real world census data.

6.1 Case Study Data Landscape

The business data landscape of the Bank of the States stores information on each processing stage of a mortgage, ranging from the initial referral, to a filtered referral, to an application and finally upon a successful application to a mortgage. As each mortgage case data flows through these states, the volumes of mortgage cases reduce; the volume of mortgage cases at each stage represents demand for the bank.

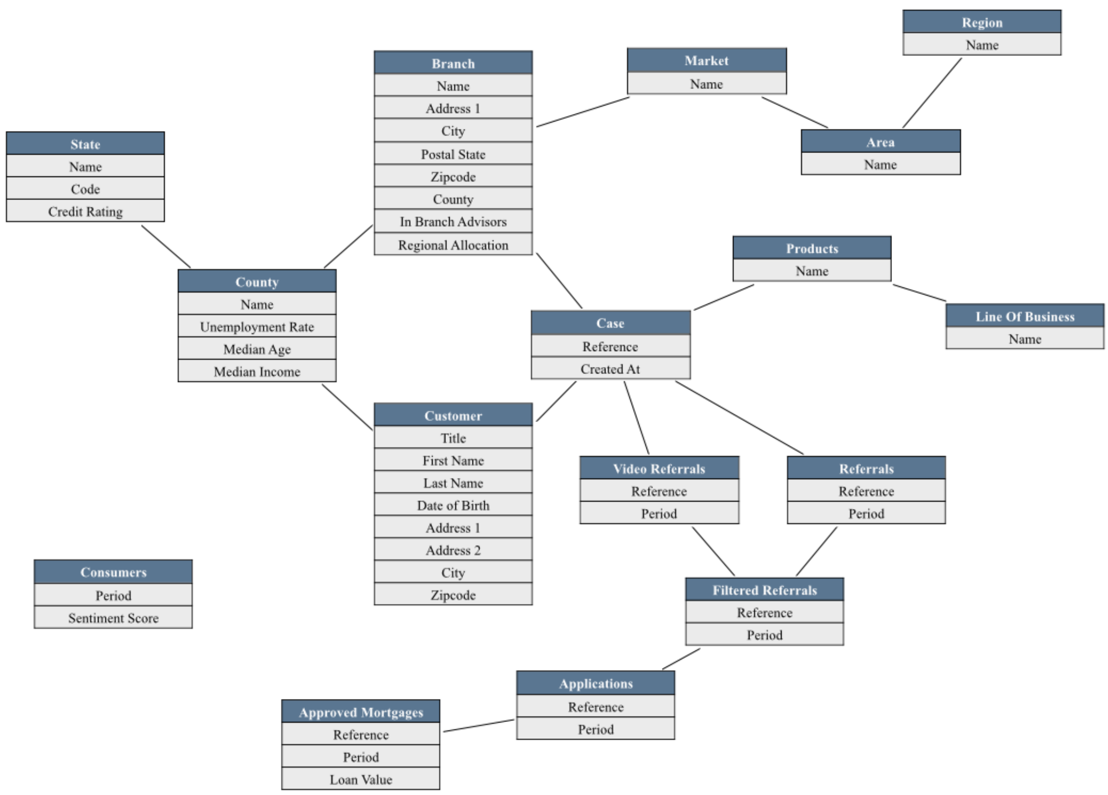


Figure 28: Bank of the States data model

Branch

This data element contains specific information about each bank branch in the Bank of the States network. Each Branch is associated with a Market in which it operates in.

Referral

The data element represents reference information for an enquiry about a mortgage application or interest in a mortgage from either an existing or potential new bank customer. Referrals can come from various sources such as an adhoc enquiry from marketing efforts or existing customers requiring new or different mortgage packages.

Filtered Referral

This data element represents the progressed state of a Referral which has passed basic checks by the bank. These checks include a suitability check or a basic affordability check by the mortgage advisers

Applications

This data element represents the progressed state of a Filtered Referral which is in the mortgage application process, and pending all affordability, security and property checks by the mortgage advisers.

Approved Mortgages

This data element represents the progressed state of a mortgage application which has passed all checks and the mortgage has been issued to the customer by the bank.

The capacity of the bank to handle the demand from customers relating to mortgage applications is provided by a number of different sources. Currently the bank has three different sources of capability for processing mortgages through the states of progression: in branch permanent mortgage advisers, regional mortgage advisers and call centre telephony mortgage advisers. In branch permanent mortgage advisers are permanently employed staff who exclusively operate from their assigned bank branch.

Regional mortgage advisers are permanently employed staff who exclusively operate in a specific region which can cover multiple bank branches. These advisers must schedule their capability time to maximise the time spent providing mortgage advice to customers in the region. Call centre telephony mortgage advisers, are permanently employed staff who provide mortgage application services to all bank branches via a telephony call centre.

The volumes of customer mortgage cases at each stage in the process is what generates demand on the banks mortgage adviser resource pool in the form of a level full time equivalent (FTE). Each mortgage case stage generates demand in different ways. Referrals and Filtered Referrals generate demand on the *in branch* and *call centre* advisers; Applications and Approved Mortgages generate demand on the all the mortgage adviser archetypes.

6.2 Case Study Causal Model

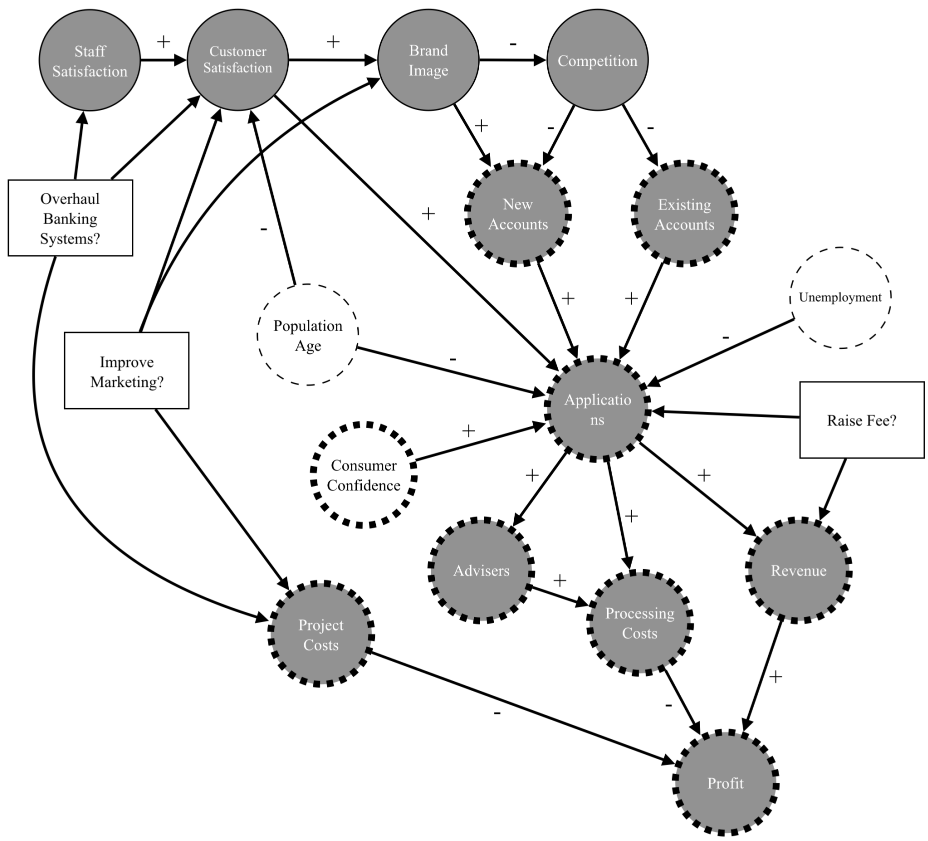


Figure 29: Bank of the States causal model

The model variables in figure 29 are designed to highlight the most significant drivers that the bank feels contribute towards mortgage applications and how these contribute toward profits for the bank. The variables shown with light dashed edges represent the social and economic influencing factors by which the bank attributes as the key causes which can affect the volumes of customers, both existing and new, with respect to customer satisfaction and mortgage applications. The heavy dashed grey variables are continuous and the non-dashed grey variables are discrete.

For the purposes of this case study, it is assumed that the example model structure is fixed and is the optimal model, whereas in reality the model structure may well evolve as the understanding of the model domain evolves and/or additional data becomes available. The fixing of the model structure aids the example with parameter value generation, as detailed in section 6.2.2. This iterative nature of refinement will occur, and in this scenario of building models with limited or no available data, the assessment of whether the model is optimal must be done by using the analysis techniques detailed in sections 6.3 and 6.4 to establish if both the parameters and model results are close to the domain expert's expectation. The subject of learning a causal model structure basically involves one of three different approaches: constraint based learning, score based learning and Bayesian model averaging. Finding the optimal causal model again involves an iterative approach, only in this instance the evaluation is based on maximising the value of the scoring function used. The subject of model structure learning from data is beyond the scope of this document, however further information can be found at [15].

The model variable *Consumer Confidence* is based upon the *Consumer Sentiment Index* score as published by the University of Michigan [16]. This score, published on a monthly basis, is a measure of consumers' confidence and attitudes towards spending and the overall business climate. Financial businesses utilise this score as a measure of judgement towards consumer levels of optimism regarding possible future spending.

The model variables *Unemployment* and *Population Age* are US County level statistics gathered from US census data [17] [18]. The *Population Age* variable is a measurement of the mean population age by US County.

There are three decision variables within the model that describe the three decisions that the bank faces with regards to overhauling the banking systems, running a marketing campaign and lastly whether to increase the mortgage fee or not. Raising the fee is a secondary decision that the bank feels it may have to introduce if it feels that the costs for performing the system upgrades, marketing and any changes in staff capacity need to be recuperated.

The final objective variable for the model is *Profit*, which of course has the overall goal of being maximised. In this model the *Profit* variable is a simple formula of *Revenue* - *Cost per Application* - *Project Costs*. The bank has budgeted a maximum spend of $1,000,000 on marketing and an estimated spend of $250,000 on a banking systems upgrade.

Additionally, figure 29 describes the general direction of influence in the causal model, as shown using the + and - signs on each influence relationship. A + indicates that the parent variable has a positive influence on its child variable, that is an increase in the parent variable value results in an increase in the child variable; the same effect is true when the parent if decreased, a decrease occurs in the child variable. A - influence indicates that the parent has a negative influence on its child variable, that is an increase in the parent variable value results in a decrease in the child variable; the inverse effect is true.

6.2.1 Decisions for the bank

As mentioned the Bank of the States has seen a decline in customer satisfaction over the previous 18 months, and has attributed this to the growing need to have an online presence for new applications and management of existing mortgages. The bank is also aware from feedback that the banking staff are frustrated with the current banking systems and find them counter-intuitive to use, the bank feels that staff satisfaction is also an important factor for customer satisfaction. This is expressed in the model with the decision to *Overhaul the Banking Systems* directly influencing both *Staff Satisfaction* and *Customer Satisfaction*, with an additional influence relationship from *Staff Satisfaction* to *Customer Satisfaction*.

The decision to run a marketing campaign can have significant benefits, as it directly influences *Customer Satisfaction* and *Brand Image*, with the latter further influencing the market share from competitors which the bank feels is an ever increasing problem currently. However, running the intended marketing campaign will be a very costly project.

Either or both of these key project decisions for the bank will address the customer satisfaction issue it faces and start to increase the mortgage applications for the bank. However, with the potential increase in applications there will be increase in the required capacity to process these them. Therefore the model variable *Advisers* is an estimated number of the Full Time Equivalent (FTE) staff headcount required to process the estimated changes to the volume of mortgage applications. The associated costs for the *Advisers* is made visible by the variable *Processing Costs*.

As shown in figure 30, the variables *Population Age* and *Unemployment* are optionally contextualised from the business data model. The contextual data for these variables are all derived from a level of granularity as set by a specific US County, which is passed to the business data model as a filtering variable. For example, the model variables *Population Age*, *Unemployment* get their contextual values from the County that the a given Branch is situated in.

The only exception to this is the causal model variable *Consumer Confidence*, which simply contextualises the model variable with the data currently available from the data model given the current month and year.

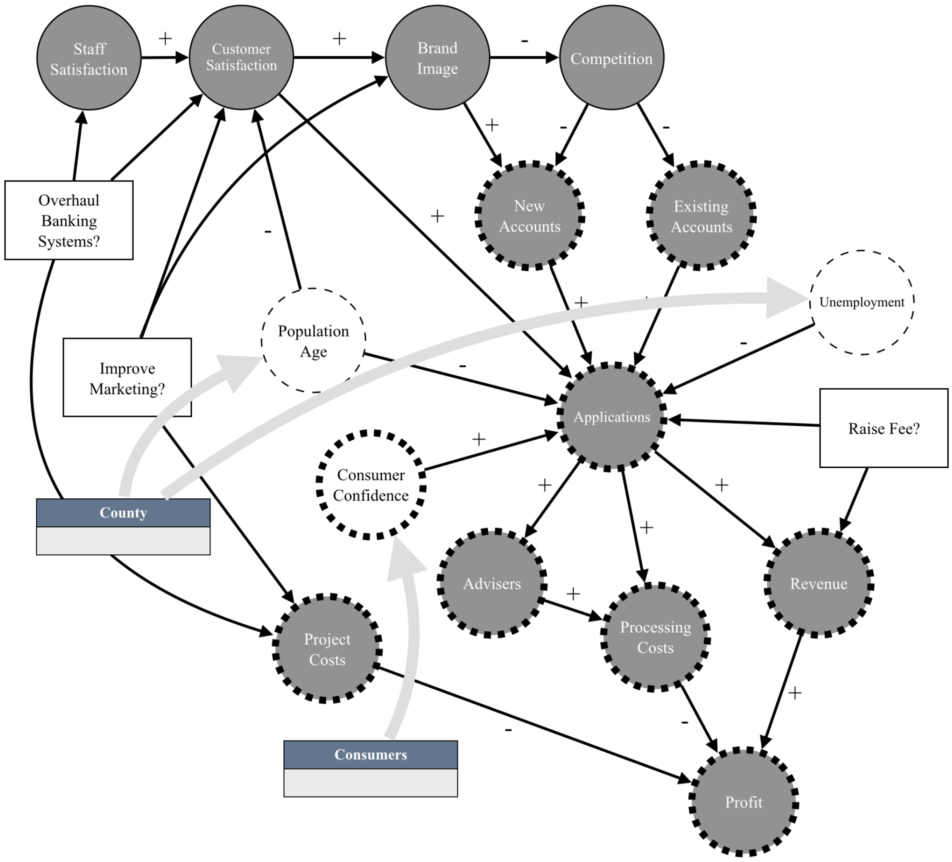
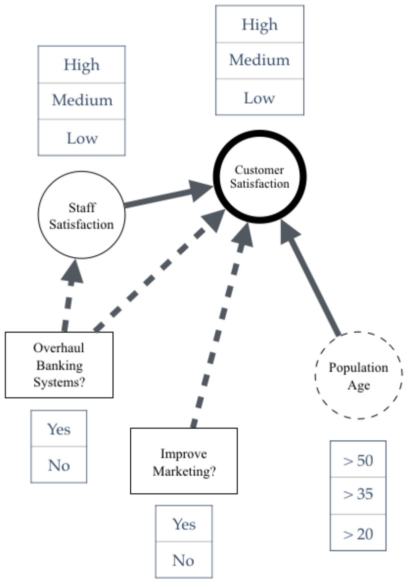


Figure 30: Bank of the States data linked causal model

6.2.2 Generated CPD Configurations

The below figures highlight the configurations created for the model variables *Customer Satisfaction*, *Brand Image* and *Competition*. These configurations are also detailed in Appendix A.3 along with the generated probabilities.

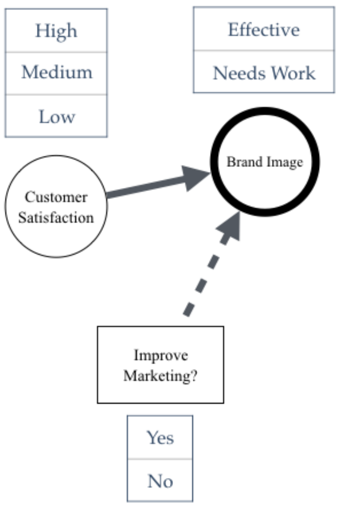


(a)

				Customer Satisfaction		
Marketing	Overhaul	Population Age	Staff Satisfaction	High	Medium	Low
Yes	Yes	> 50	Medium			VL
Yes	Yes	> 35	Medium	VL		
Yes	Yes	> 20	High	VL		
Yes	No	> 50	High	FL		
Yes	No	> 35	Medium		FL	
Yes	No	> 20	High	VL		
No	Yes	> 50	Medium			VL
No	Yes	> 35	Medium		VL	
No	Yes	> 20	High	FL		
No	No	> 50	High		VL	
No	No	> 35	Medium			FL
No	No	> 20	Low			VL

(b)

Figure 31: Customer Satisfaction CPD configuration.

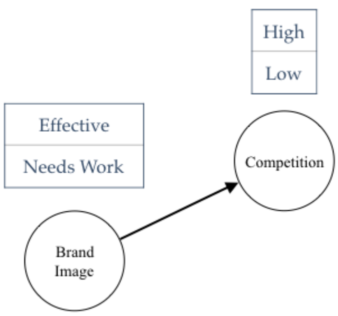


(a)

		Brand Image	
Marketing?	Customer Satisfaction	Effective	Needs Work
Yes	High	**AC**	
No	Low		FL

(b)

Figure 32: Brand Image CPD configuration.



(a)

	Competition	
Brand Image	High	Low
Effective	**VL**	

(b)

Figure 33: Competition CPD configuration.

6.3 Sensitivity Analysis

When designing models for a business and subsequently completing the conditional probabilities, sensitivity analysis should be performed on model to establish which of the model nodes are more or less tolerant to error with respect to the outcome model variable.

Sensitivity analysis should be performed on the Causal Bayesian Network to identify the most influential parameters. And finally more accurate values of those identified probability parameters can be obtained by a more careful assessment. The process of iteratively performing sensitivity analysis and refining those influential parameters until a satisfactory behaviour of the Causal Bayesian Network is achieved is recommended to be done until the cost of further elicitation outweighs the benefits of higher accuracy, or until higher accuracy can no longer be obtained due to a lack of knowledge. In this iterative procedure, an expert can focus his or her attention on the probabilities to which the network's behaviour shows highest sensitivity. Those less influential parameters can be left with crude estimates.

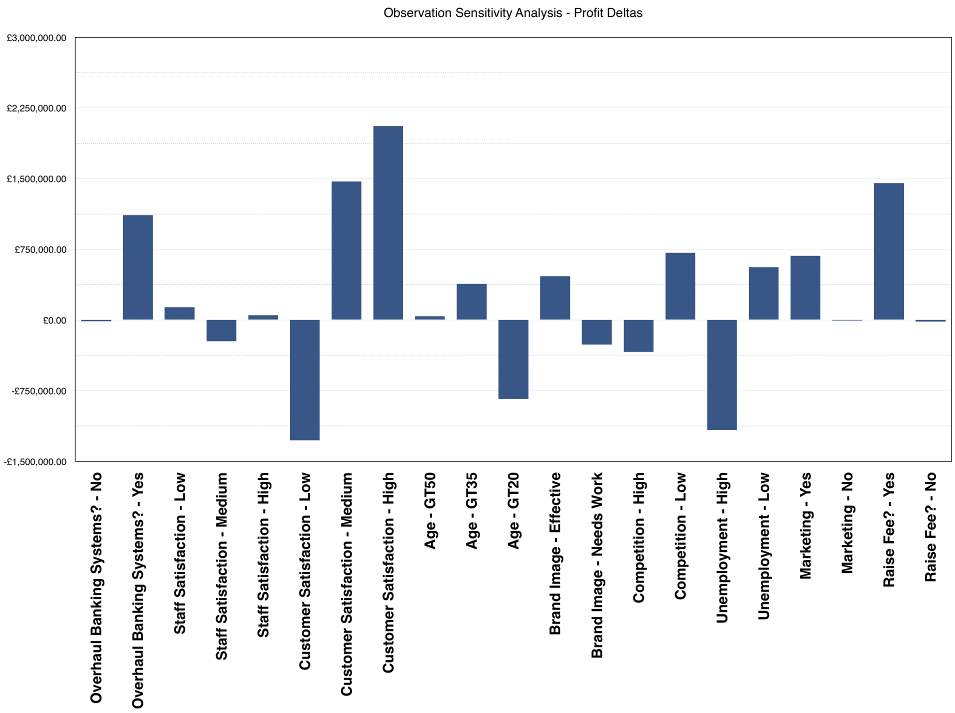


Figure 34: Observation based analysis on Profit

Figure 34 shows the results of observing each discrete variable's states individually and then recording the delta value on the outcome variable "Profit". The most influential variable in the model is the "Customer Satisfaction" variable, which makes sense for a retail focused business. This gives some insight into which variables in the model yield the largest and smallest effect on the outcome variable.

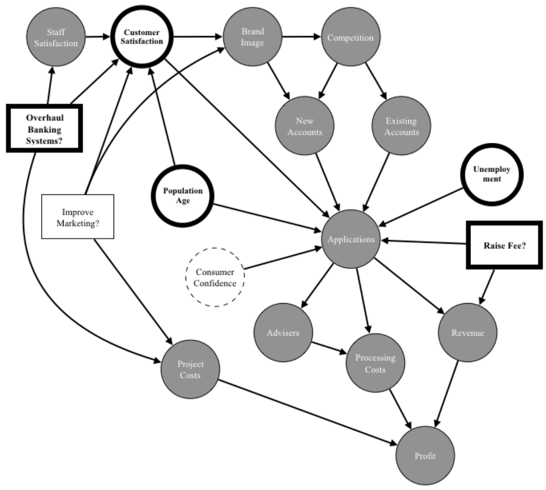


Figure 35: Case study model with the most influential variables highlighted.

Figure 35 shows the model with the variables that result in the biggest value changes for the outcome variable "Profit". These variables provide a good starting point for sensitivity analysis techniques [19] to be performed to tune the variables, as they are less tolerant to errors. However, in this specific case, due to the *Population Age* and *Unemployment* variables being derived from actual census data its safe to disregard these from sensitivity analysis, leaving just *Customer Satisfaction*.

6.4 Parameter Validation

When a business creates a Causal Model using a domain expert as the primary source, and uses the method described in this document for generating certain discrete variables, there is a need to ensure that the generated variables model the influences valid enough. It is necessary to perform some validation for these generated variables, traditionally if the business has access to observational data then there are scoring techniques, such as the Brier score [20], to measure how far from the observation the probability is [21], with values closer to zero being the most optimal score. In the situation catered for here, the business does not have any observational data, for example in the case study the bank does not have any data related to the variable *Customer Satisfaction*. These more behavioural model variables are difficult to validate against given this lack of a comparison. A Brier score could be specified by using expected values which are elicited from the domain expert, however careful management of this would need to be done so that unwanted biases are not introduced into the evaluation. The validation method would be to calculate the Causal Bayesian Network in an iterative manner for a scenario which is aligned to past observational data for the outcome model variables, making observations in a specific model variable and comparing the Brier score for the outcome variable with the expectation of the domain expert.

For example, when validating the variable *Customer Satisfaction*, the case study model can be calculated for previous time periods for which the bank has operational data relating to the *Applications* variable. The model is calculated using any other observational data available in this time period, in this case any of the variables *Population Age*, *Consumer Confidence* or *Unemployment Rate*. Due to the lack of any observational data for *Customer Satisfaction* the time period selection should be based on a period which the domain expert has any other available indicators for the potential state of *Customer Satisfaction* at that time, this could include researching any economic factors present in this period which may give some indication of the state of *Customer Satisfaction*. The model should be calculated with the observations set and a Brier score calculated for the variable being validated. The domain expert would then need to make an assessment of how tolerant of the score they should be in order to inform any manual adjustments in the CPD parameter values of the variable being validated.

Figure 36: Brier score calculated for Customer Satisfaction for existing observations of Applications with domain expert expectations for Customer Satisfaction of Medium and High

Figure 36 shows the Brier scores of a sample set of observed values for the *Applications* variable which the bank has available for prior time periods. Using the equation $BS = \frac{1}{N}\sum_{N}^{t=1}(f_t - o_t)^2$ to calculate the Brier score given $N = 3$, the domain expert expected observation o_t for *Customer Satisfaction* and the probability of *Customer Satisfaction* at t for the observation of *Applications* at t, each series on the chart represent the Brier score calculated given the domain experts expected observation for *Customer Satisfaction*. We can speculate that for the observed values of *Applications* between 1654 and 1814, the domain experts expected value of *Medium* for *Customer Satisfaction* for the most part matches the expectation. However, there is a peculiarity for the range 1684 to 1704, whereby the value of *High* has a better score, this will need further analysis in the model to establish if other factors are contributing to this anomaly. Using the observed values for *Applications* above 1814 suggest a *Customer Satisfaction* value of *High* is more suited.

7 Conclusions

Using the methods described in this thesis to generate discrete conditional probability distributions, enables a domain expert to quickly get a hypothesis model up and running and potentially ready to perform decision making upon. The ability for a business to utilise a Causal Bayesian Network for decision making is key as they allow a domain expert to gain visibility and a deeper understanding of the business processes. Further, from this better understanding, business process changes and the probable impacts can be explored and analysed. Many businesses gather various volumes of data, from high level process / metric data to vast quantities of more social / diagnostic data. At the recent ICCRTS (International Command and Control Research and Technology Symposium) 2016, the recurring problem that companies had was that they have too much data now, whereas previously they had the problem that they had not enough data. At the conference there was a strong feeling that simply gathering all of this data was not enough, and the ability to use it to perform analysis on the real methodology of a business was much more important. The prospect of making sense and modelling business processes without necessarily requiring the need to perform data-mining or various data aggregation techniques is very desirable. When dealing with observation data, it does not answer the question of how/why the data is the value it is. It is the ability of the Causal Bayesian Network to model these values and by modelling the values gain an understanding into what drives the values. These causal drivers may well be influencing factors within or external to a business, and as such perhaps are not available in data from internal systems. For example, economic / social factors which could be influencing in a more behavioural way, which is not explicitly captured in data. In the use case, the ability to model the social and economic factors based upon domain expert knowledge means that the modelled outcome can factor in these influences. If the bank simply gathered this data and performed analysis techniques such as a Naive Bayes algorithm, whilst the degree of movement from one data variable to the correlated data is accurate in so far as the data can provide, it would not allow an analyst to encode domain expert knowledge.

The method of generating conditional probability distributions is therefore a key requirement, and using constraint satisfaction problem approach means that it is very fast for a domain expert to create candidate causal models. As with any Causal Bayesian Network, whether it is generated from data or manually created by domain experts, there is always a need to validate the model to ensure that the outcome values are close enough to either any observational values available or to expectations. The iterative process of model creation, parameter generation and validation should ideally be fast so that the creation of a functioning Causal Bayesian Network is not a prohibitive task.

8 Further Research

The method for generating conditional probabilities described here could be further extended with some research on Probability Assessment with Maximum Entropy [22]. In this research the conditional probabilities are generated to closely match with an expected result on model variables. Therefore if a domain expert already has an expectation of what the result should be given conditions on influencing variables, the conditional probabilities generated for those influencing variables could be made much closer to the expectation. With the generation method described here, when a solution to the constraint problem is found the conditional probabilities are valid, are selected from a finite set of possibilities determined by your level of discretisation. As such in any case of generated probabilities the sensitivity analysis must be performed.

Ultimately any probabilities generation technique should try to maintain an unbiased nature. The entropy based generation could suffer from domain expert biases for the expectation that they could specify.

Another potential application for conditional probability generation could be in the use of Credal Networks [23], which is a move towards an imprecise method of computing probabilities. Credal networks allow a result to be expressed in a more imprecise manner, such as "X is more probable than Y", rather than in an uncertainty model like Causal Bayesian Networks whereby a result is expressed as "X is three times more probable than Y". Using the constraint defining configurations defined here, the constraint solver could output the possible ranges available during the solution findings, rather than selecting a specific probability value within the range.

A Appendices

A.1 Probability Word Estimator Survey

Survey Sceanrios		
Question	Word of Probability	Scenario
1	Impossible	Matt places 6 RED balls into an empty bag, he then passes the bag to you and asks you to remove a ball from the bag. It is IMPOSSIBLE that you will remove a BLUE ball from the bag.
2	Very Unlikely	You are visiting London and it is Christmas Day, it is VERY UNLIKELY that it will be snowing.
3	Fairly Unlikely	It is currently Summer in the UK, and the weather has been dry today, it is FAIRLY UNLIKELY that it will rain tomorrow.
4	Fifty-Fifty	Geoff tosses a coin, and openly states beforehand that he thinks it will land with he heads side facing up, it is FIFTY-FIFTY that this outcome will be true.
5	Fairly Likely	Carol leaves her house, it is dark outside, it is FAIRLY LIKELY late in the evening.
6	Very Likely	Steve is waiting to undertake a public speech to a large audience, it is VERY LIKELY that Steve is nervous.
7	Almost Certain	Mark is on the 5th floor of a building, he opens the window and drops a glass cup onto a tarmac road below, it is ALMOST CERTAIN that the glass will break.

Table 14: Words of Probability Survey Scenarios

Survey Results						
Question	Min	Max	Mean	Std Dev	Variance	Count
1	0.00	100.00	61.13	48.29	2331.83	63
2	2.00	100.00	28.80	29.91	894.74	108
3	1.00	87.00	35.93	17.68	312.52	104
4	49.00	100.00	60.26	19.72	388.95	108
5	24.00	100.00	67.27	14.24	202.68	108
6	9.00	100.00	77.79	17.90	320.26	107
7	50.00	100.00	94.52	7.71	59.50	112

Table 15: Words of Probability Survey Results Summary

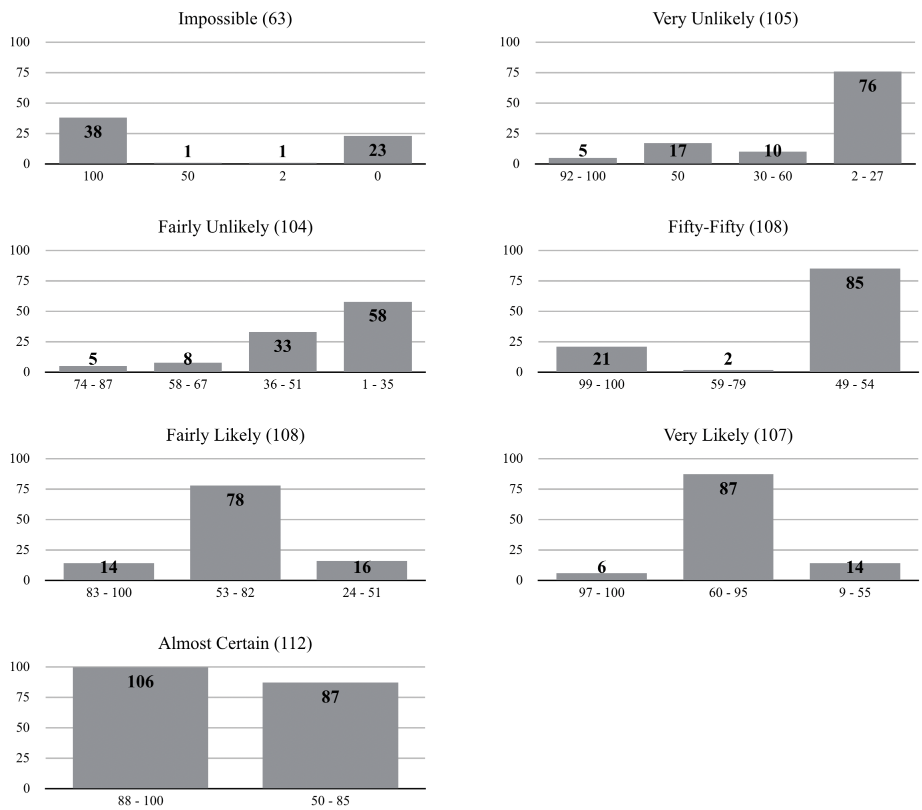


Figure 37: Aggregated survey response counts for each word of probability.

The survey was produced and presented using Qualtrics [24] and all survey results are stored securely by Qualtrics.

A.2 Generated Probabilities Examples

The below figures highlight the min, max and mean values for the domain ranges during constraint solving for a CPD. Each row of figures shows these ranges, firstly for each CPD cell in the row of the influence block which contains the anchor cell. Secondly, the ranges are shown for each CPD cell of the non-anchor neighbouring rows in an influence block. The sample Bayesian network fragment and CPD shape used for constraint generation is structured as shown in Fig. 38

		C			
A	B	1	2	3	4
1	1	**Anchor**			
1	2	Neighbour Anchor			
1	3	Neighbour Anchor			
2	1			Neighbour Anchor	
2	2			**Anchor**	
2	3			Neighbour Anchor	
3	1			Neighbour Anchor	
3	2			Neighbour Anchor	
3	3			**Anchor**	

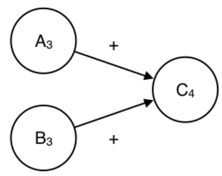
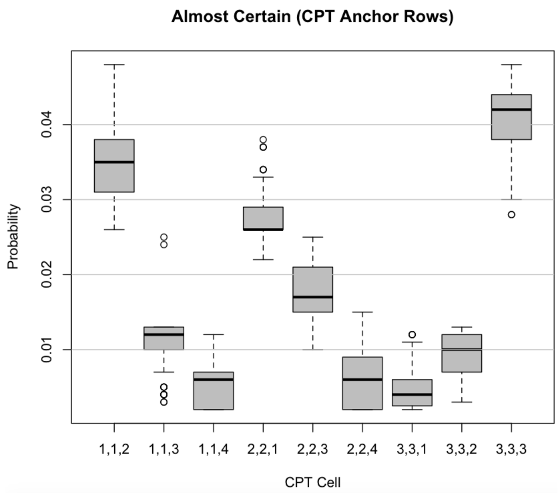
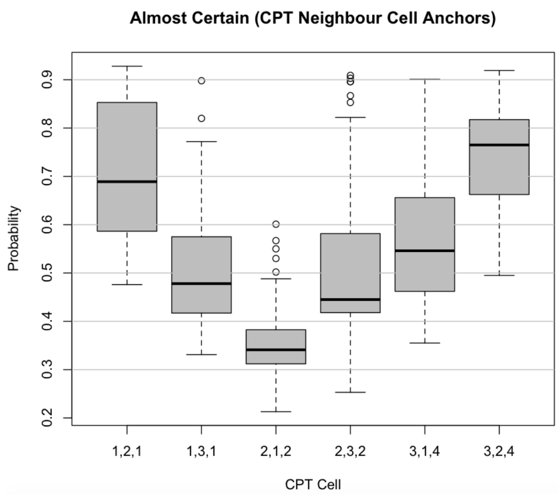
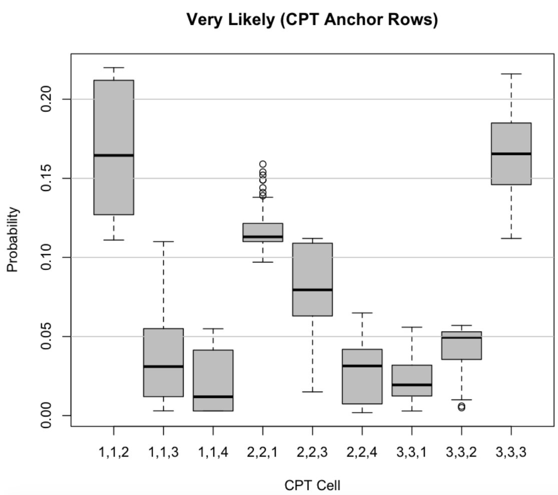
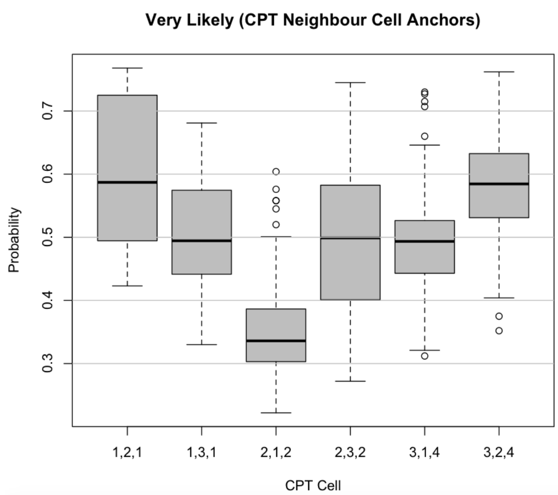


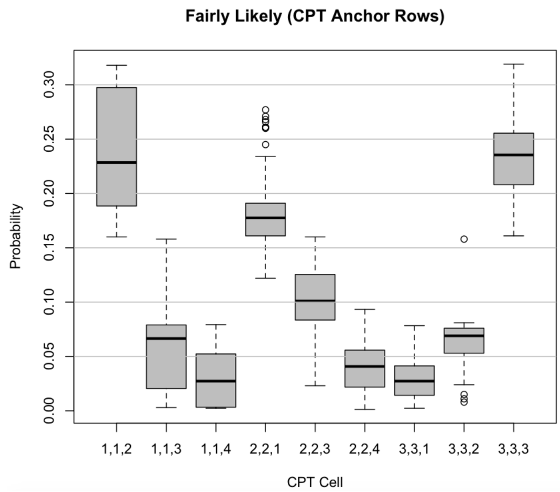
Figure 38

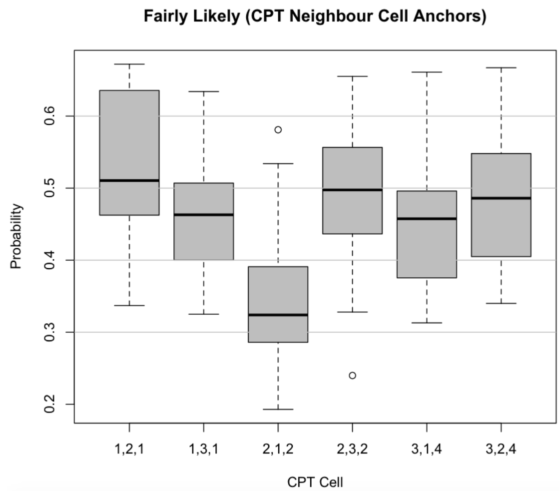


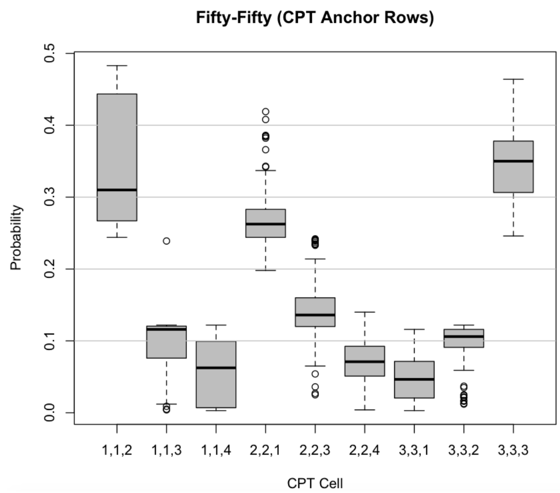


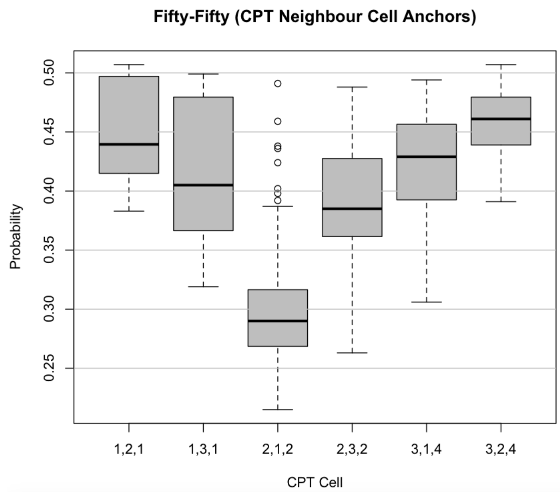












A.3 Case Study Causal Model Probabilities With Generated CPD Configurations

The case study model CPD parameters are a mix of generated probabilities, domain expert assessments and values analysed from business and/or census data. Listed below are all the CPD values and configurations for each discrete or continuous model variable, along with a description of how the values where created in the case of data and domain expert derived values.

A.3.1 Domain Expert Assessments

Staff Satisfaction		
Overhaul?	State	Probability
Yes	High	0.4
Yes	Medium	0.4
Yes	Low	0.2
No	High	0.1
No	Medium	0.35
No	Low	0.55

Table 16: Staff Satisfaction probabilities

Table 17: Applications probabilities

Applications							
				Beta			
RF?	CS	U	Age	EA	NA	CC	Variance
No	High	High	>50	0.03	0.03	1.5	1
No	High	High	>35	0.03	0.06	1.5	1
No	High	High	>20	0.03	0.06	1.5	1
No	High	Low	>50	0.06	0.03	1.5	1
No	High	Low	>35	0.09	0.15	1.5	1
No	High	Low	>20	0.06	0.012	1.5	1
No	Medium	High	>50	0.03	0.03	1.5	1
No	Medium	High	>35	0.024	0.027	1.5	1
No	Medium	High	>20	0.018	0.015	1.5	1
No	Medium	Low	>50	0.03	0.03	1.5	1
No	Medium	Low	>35	0.09	0.12	1.5	1
No	Medium	Low	>20	0.03	0.09	1.5	1
No	Low	High	>50	0.03	0.03	1.5	1
No	Low	High	>35	0.012	0.015	1.5	1
No	Low	High	>20	0.009	0.006	1.5	1
No	Low	Low	>50	0.024	0.024	1.5	1
No	Low	Low	>35	0.018	0.018	1.5	1
No	Low	Low	>20	0.018	0.018	1.5	1
Yes	High	High	>50	0.027	0.027	1.5	1

Table 17: Applications probabilities

Applications							
				Beta			
RF?	CS	U	Age	EA	NA	CC	Variance
Yes	High	High	>35	0.027	0.054	1.5	1
Yes	High	High	>20	0.027	0.027	1.5	1
Yes	High	Low	>50	0.054	0.027	1.5	1
Yes	High	Low	>35	0.081	0.135	1.5	1
Yes	High	Low	>20	0.054	0.108	1.5	1
Yes	Medium	High	>50	0.027	0.027	1.5	1
Yes	Medium	High	>35	0.0216	0.0243	1.5	1
Yes	Medium	High	>20	0.0162	0.0135	1.5	1
Yes	Medium	Low	>50	0.027	0.027	1.5	1
Yes	Medium	Low	>35	0.081	0.108	1.5	1
Yes	Medium	Low	>20	0.027	0.081	1.5	1
Yes	Low	High	>50	0.027	0.027	1.5	1
Yes	Low	High	>35	0.0108	0.0135	1.5	1
Yes	Low	High	>20	0.0081	0.0054	1.5	1
Yes	Low	Low	>50	0.0216	0.0216	1.5	1
Yes	Low	Low	>35	0.0162	0.0162	1.5	1
Yes	Low	Low	>20	0.0162	0.0162	1.5	1

Project Costs			
Overhaul?	Marketing?	Mean	Variance
Yes	Yes	1250000	1
Yes	No	250000	1
No	Yes	1000000	1
No	No	0	1

Table 18: Project Costs probabilities

A.3.2 Data Driven Assessments

The *Unemployment Rate* variable is derived from the US census unemployment rates for every county in each US state [18]. The data was discretised into a High and Low state based upon an unemployment boundary threshold of 6.9 percent over data averaged over a one year period in 2016 for 3196 US counties.

Unemployment	
State	Probability
High	0.1257
Low	0.8743

Table 19: Unemployment probabilities, extracted from Census data

The *Population Age* variable is derived from the US census average population age for each US state as of 2016 [17]. The data was discretised into three states that the bank felt best represented its loan application demographics groups. Each discrete state value boundary is based on the average population age being between 20 and 35, between 35 and 50 and over 50.

Age	
State	Probability
>50	0.05
>35	0.65
>20	0.3

Table 20: Age probabilities

The *Consumer Confidence* variable is derived from the average of the Consumer Confidence Score from the University of Michigan [16] for 2016.

Consumer Confidence	
Mean	Variance
91.84	8.5

Table 21: Consumer Confidence probabilities

The *Existing Accounts* variable is derived from account data that the *Bank of the States* has stored over the past 10 years. Analysis was performed by the bank's data analysts over the dataset guided by the domain expert as to when the bank had High or Low competition from other banks over the 10 year period of the data.

Existing Accounts		
Competition	Mean	Variance
High	35000	54.77
Low	50000	31.62

Table 22: Existing Accounts probabilities

The *New Accounts* variable is based on data that the *Bank of the States* has stored over the past 10 years on new account applications by month. The data analysts were guided by the domain expert on the level of competition and how effective they felt the brand image was in each monthly period.

New Accounts			
Brand Image	Competition	Mean	Variance
Effective	High	5200	15.88
Effective	Low	8500	12.25
Needs Work	High	2700	8.66
Needs Work	Low	4000	12.25

Table 23: New Accounts probabilities

The *Advisors* continuous variable is derived from Human Resources time allocation data that the *Bank of the States* has available. On a daily basis each bank employee must complete a timesheet breakdown of their work hours broken down by cost codes, therefore the data analysts were able to extract how much FTE (Full Time Equivalent) effort a bank loan advisor takes to process loan applications. From this computed work rate the analysts are able to calculate, for a single loan application, the percentage time taken per FTE.

Advisers		
Applications	Mean	Variance
0.02	1	1

Table 24: Advisers probabilities

The *Processing Costs* continuous variable is derived from from analysing accounts data, and extracting how much it costs the bank to process a single application in terms of both the cost of each Advisor as well as administration costs.

Processing Costs		
Applications	Advisers	Variance
150	10000	1

Table 25: Processing Costs probabilities

The *Revenue* continuous variable is derived from accounts data at the bank, taken over the past 10 years. The data split by the decision *Raise Fee?* is differentiated based on computing how much more revenue can be made by increasing the loan application fee by an amount suggested by the domain expert on behalf of the business.

Revenue		
Raise Fee?	Applications	Variance
Yes	2900	1
No	1800	1

Table 26: Revenue probabilities

The *Profit* continuous variable is a simple linear distribution which describes the positive / negative nature of the influences of *Processing Costs*, *Project Costs* and *Revenue*.

Profit			
Processing Costs	Revenue	Project Costs	Variance
-1	1	-1	1

Table 27: Profit probabilities

A.3.3 Generated Probabilities

Table 28: Customer Satisfaction probabilities

Customer Satisfaction					
Marketing?	Overhaul?	Age	Staff Satisfaction	State	Probability
Yes	Yes	>50	High	High	0.133
Yes	Yes	>50	High	Medium	0.319
Yes	Yes	>50	High	Low	0.548
Yes	Yes	>50	Medium	High	0.0669
Yes	Yes	>50	Medium	Medium	0.16
Yes	Yes	>50	Medium	Low	0.7731
Yes	Yes	>50	Low	High	0.101
Yes	Yes	>50	Low	Medium	0.391
Yes	Yes	>50	Low	Low	0.508
Yes	Yes	>35	High	High	0.67
Yes	Yes	>35	High	Medium	0.25
Yes	Yes	>35	High	Low	0.08
Yes	Yes	>35	Medium	High	0.1539
Yes	Yes	>35	Medium	Medium	0.7731
Yes	Yes	>35	Medium	Low	0.073
Yes	Yes	>35	Low	High	0.175
Yes	Yes	>35	Low	Medium	0.209
Yes	Yes	>35	Low	Low	0.616
Yes	Yes	>20	High	High	0.7731
Yes	Yes	>20	High	Medium	0.19
Yes	Yes	>20	High	Low	0.0369
Yes	Yes	>20	Medium	High	0.686
Yes	Yes	>20	Medium	Medium	0.277
Yes	Yes	>20	Medium	Low	0.037
Yes	Yes	>20	Low	High	0.535

Table 28: Customer Satisfaction probabilities

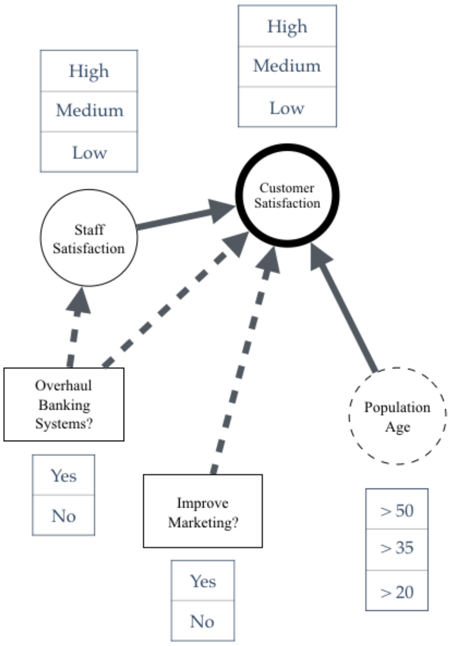
Customer Satisfaction					
Marketing?	Overhaul?	Age	Staff Satisfaction	State	Probability
Yes	Yes	>20	Low	Medium	0.452
Yes	Yes	>20	Low	Low	0.013
Yes	No	>50	High	High	0.6757
Yes	No	>50	High	Medium	0.289
Yes	No	>50	High	Low	0.0353
Yes	No	>50	Medium	High	0.634
Yes	No	>50	Medium	Medium	0.32
Yes	No	>50	Medium	Low	0.046
Yes	No	>50	Low	High	0.452
Yes	No	>50	Low	Medium	0.375
Yes	No	>50	Low	Low	0.173
Yes	No	>35	High	High	0.242
Yes	No	>35	High	Medium	0.574
Yes	No	>35	High	Low	0.184
Yes	No	>35	Medium	High	0.2833
Yes	No	>35	Medium	Medium	0.6757
Yes	No	>35	Medium	Low	0.041
Yes	No	>35	Low	High	0.588
Yes	No	>35	Low	Medium	0.216
Yes	No	>35	Low	Low	0.196
Yes	No	>20	High	High	0.6757
Yes	No	>20	High	Medium	0.188
Yes	No	>20	High	Low	0.1363
Yes	No	>20	Medium	High	0.558
Yes	No	>20	Medium	Medium	0.26
Yes	No	>20	Medium	Low	0.182
Yes	No	>20	Low	High	0.41
Yes	No	>20	Low	Medium	0.309
Yes	No	>20	Low	Low	0.281
No	Yes	>50	High	High	0.133
No	Yes	>50	High	Medium	0.319
No	Yes	>50	High	Low	0.548

Table 28: Customer Satisfaction probabilities

Customer Satisfaction					
Marketing?	Overhaul?	Age	Staff Satisfaction	State	Probability
No	Yes	>50	Medium	High	0.0669
No	Yes	>50	Medium	Medium	0.16
No	Yes	>50	Medium	Low	0.7731
No	Yes	>50	Low	High	0.101
No	Yes	>50	Low	Medium	0.391
No	Yes	>50	Low	Low	0.508
No	Yes	>35	High	High	0.67
No	Yes	>35	High	Medium	0.25
No	Yes	>35	High	Low	0.08
No	Yes	>35	Medium	High	0.1539
No	Yes	>35	Medium	Medium	0.7731
No	Yes	>35	Medium	Low	0.073
No	Yes	>35	Low	High	0.175
No	Yes	>35	Low	Medium	0.209
No	Yes	>35	Low	Low	0.616
No	Yes	>20	High	High	0.7731
No	Yes	>20	High	Medium	0.19
No	Yes	>20	High	Low	0.0369
No	Yes	>20	Medium	High	0.686
No	Yes	>20	Medium	Medium	0.277
No	Yes	>20	Medium	Low	0.037
No	Yes	>20	Low	High	0.535
No	Yes	>20	Low	Medium	0.452
No	Yes	>20	Low	Low	0.013
No	No	>50	High	High	0.1809
No	No	>50	High	Medium	0.7731
No	No	>50	High	Low	0.046
No	No	>50	Medium	High	0.343
No	No	>50	Medium	Medium	0.551
No	No	>50	Medium	Low	0.106
No	No	>50	Low	High	0.428
No	No	>50	Low	Medium	0.304

Table 28: Customer Satisfaction probabilities

Customer Satisfaction					
Marketing?	Overhaul?	Age	Staff Satisfaction	State	Probability
No	No	>50	Low	Low	0.268
No	No	>35	High	High	0.073
No	No	>35	High	Medium	0.362
No	No	>35	High	Low	0.565
No	No	>35	Medium	High	0.0683
No	No	>35	Medium	Medium	0.256
No	No	>35	Medium	Low	0.6757
No	No	>35	Low	High	0.132
No	No	>35	Low	Medium	0.395
No	No	>35	Low	Low	0.473
No	No	>20	High	High	0.101
No	No	>20	High	Medium	0.391
No	No	>20	High	Low	0.508
No	No	>20	Medium	High	0.133
No	No	>20	Medium	Medium	0.319
No	No	>20	Medium	Low	0.548
No	No	>20	Low	High	0.0669
No	No	>20	Low	Medium	0.16
No	No	>20	Low	Low	0.7731



(a)

				Customer Satisfaction		
Marketing	Overhaul	Population Age	Staff Satisfaction	High	Medium	Low
Yes	Yes	> 50	Medium			VL
Yes	Yes	> 35	Medium	VL		
Yes	Yes	> 20	High	VL		
Yes	No	> 50	High	FL		
Yes	No	> 35	Medium		FL	
Yes	No	> 20	High	VL		
No	Yes	> 50	Medium			VL
No	Yes	> 35	Medium		VL	
No	Yes	> 20	High	FL		
No	No	> 50	High		VL	
No	No	> 35	Medium			FL
No	No	> 20	Low			VL

(b)

Figure 43: Customer Satisfaction CPD configuration.

Brand Image			
Marketing?	Customer Satisfaction	State	Probability
Yes	High	Effective	0.974
Yes	High	Needs Work	0.026
Yes	Medium	Effective	0.702
Yes	Medium	Needs Work	0.298
Yes	Low	Effective	0.522
Yes	Low	Needs Work	0.478
No	High	Effective	0.51
No	High	Needs Work	0.49
No	Medium	Effective	0.35
No	Medium	Needs Work	0.65
No	Low	Effective	0.3243
No	Low	Needs Work	0.6757

Table 29: Brand Image probabilities

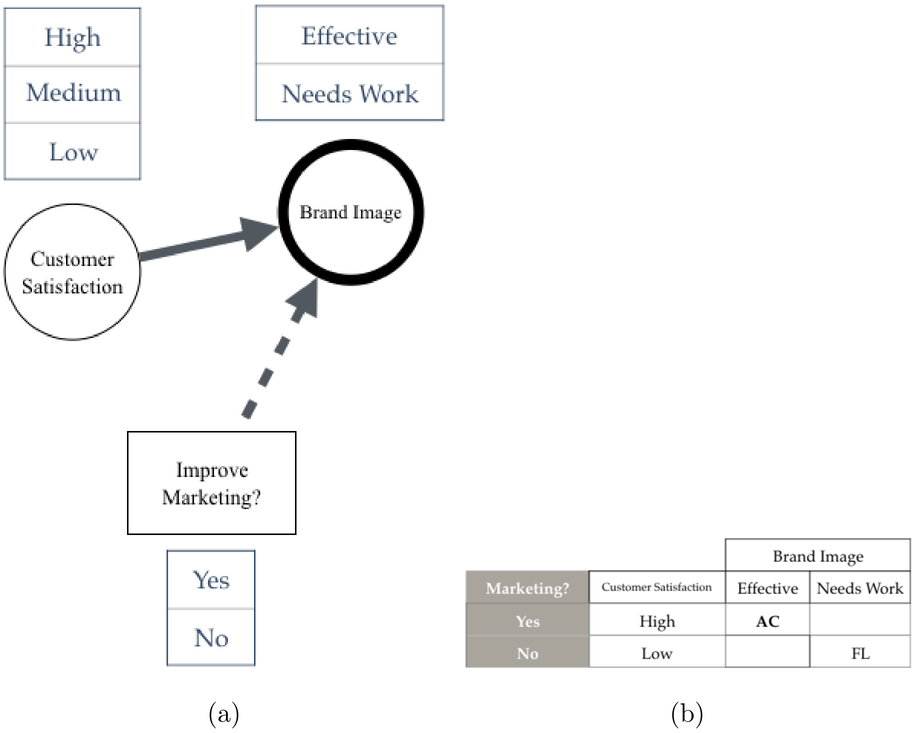


Figure 44: Brand Image CPD configuration.

Competition		
Brand Image	State	Probability
Effective	High	0.51
Needs Work	Low	0.49
Effective	High	0.7731
Needs Work	Low	0.2269

Table 30: Competition probabilities

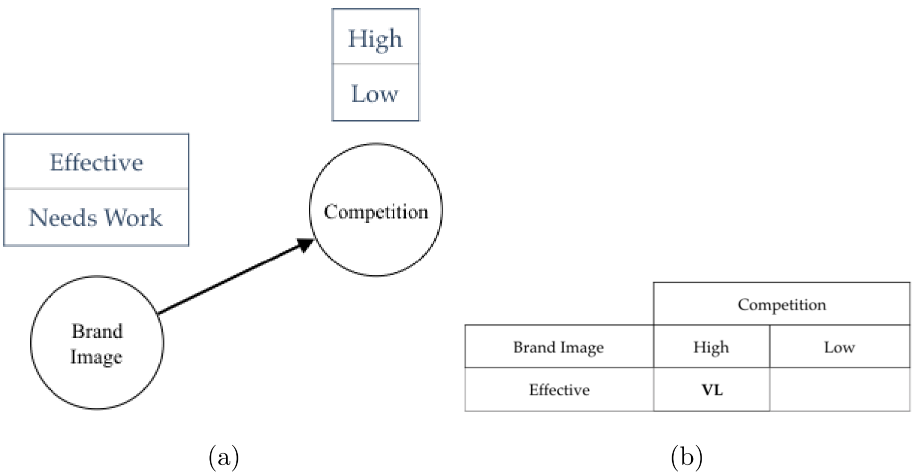


Figure 45: Competition CPD configuration.

References

[1] Marc Kerremans. “Market Guide for Enterprise Business Process Analysis”. In: (2015).

[2] Michael P. Wellman. “Fundamental concepts of qualitative probabilistic networks. Artificial Intelligence Volume 44, Issue 3”. In: (August 1990), pp. 257–303.

[3] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

[4] Charles Prud’homme, Jean-Guillaume Fages, and Xavier Lorca. *Choco Documentation*. TASC, INRIA Rennes, LINA CNRS UMR 6241, COSLING S.A.S. 2016. URL: http://www.choco-solver.org.

[5] Nir Freidman Daphne Koller. *Probabilistic Graphical Models*. MIT Press, 2009, pp. 16, 23–24.

[6] URL: https://en.wikipedia.org/wiki/Conditional_independence.

[7] Judea Pearl. *Causality: Models, Reasoning and Influence*. Cambridge University Press, 2009.

[8] Robert G. Cowell et al. *Probabilistic Networks and Expert Systems*. Springer Science + Business Media, 2007.

[9] F. Jensen S. L. Lauritzen. “Stable local computation with conditional Gaussian distributions. Statistics and Computing”. In: (2001), pp. 191–203.

[10] Amos Tversky Daniel Kahneman Paul Slovic. *Judgment under uncertainty : heuristics and biases*. Cambridge University Press, 1982.

[11] R.T. Clemen and R.L. Winkler. “Combining Probability Distributions From Experts in Risk Analysis”. In: (1999).

[12] Robert M Hamm. “Selection of verbal probabilities: A solution for some problems of verbal probability expression, 48”. In: (1991), pp. 193–223.

[13] George A. Miller. “The Magical Number Seven, Plus or Minus Two: Some Limits On Our Capacity For Information Processing. Psychology Review, 63”. In: (1956), pp. 81–87.

[14] Peter van Beek Francesca Rossi and Toby Walsh. *Handbook of Constraint Programming*. Elsevier, 2006, pp. 16, 30–31.

[15] Nir Freidman Daphne Koller. *Probabilistic Graphical Models*. MIT Press, 2009, pp. 697–995.

[16] URL: https://data.sca.isr.umich.edu/papers.php.

[17] URL: https://www.census.gov/en.html.

[18] URL: https://data.bls.gov/map/MapToolServlet?survey=la&map=county&seasonal=u.

[19] Hei Chan. "Sensitivity Analysis of Probabilistic Graphical Models". In: (2005).

[20] Kate Bull David J Spiegelhalter Rodney C G Franklin. "Assessment, criticism and improvement of imprecise subjective probabilities for a medical expert system". In: (1990).

[21] Silja Renooij. *Forecast Verification and the Uncertain Truth.* URL: http://www.staff.science.uu.nl/~renoo101/Prof/PDF/Conf/newscore.pdf.

[22] Tom Heskes Wim Wiegerinck. *Probability Assessment with Maximum Entropy in Bayesian Networks.* URL: http://www.snn.ru.nl/v2/serve.php?doc=Wiegerinck_Heskes_Interface01.pdf.

[23] Alessio Benavoli Denis Deratani Mauá Cassio P. de Campos and Alessandro Antonucci. "Probabilistic Inference in Credal Networks: New Complexity Results". In: (2014).

[24] *Qualtrics*. Qualtrics. Provo, Utah, USA, 2017. URL: http://www.qualtrics.com.

[25] URL: http://www.governing.com/gov-data/economy-finance/average-credit-score-by-state.html.

[26] Jon Holt. *A Pragmatic Guide to Business Process Modelling.* The British Computer Society, 2006.

[27] Sherman Kent. *Words of Estimative Probability.* URL: https://www.cia.gov/library/center-for-the-study-of-intelligence/csi-publications/books-and-monographs/sherman-kent-and-the-board-of-national-estimates-collected-essays/6words.html.

[28] Krzysztof R. Apt. *Principles of Constraint Programming.* Cambridge University Press, 2003.