BELIEF CHANGE IN PROBABILISTIC KNOWLEDGE REPRESENTATIONS FOR OPEN AND DYNAMIC COMPUTING ENVIRONMENTS

A Thesis

Submitted to the Faculty of Science and Agriculture

University of Zululand

by

Edgar Jembere

In Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

in

Computer Science

January, 2020

University of Zululand

Republic of South Africa

Promotors: Prof S. S. Xulu, and Prof. M. O. Adigun

DECLARATION

This dissertation represents the author's own research work and has not been submitted in any form to other tertiary education for another degree or diploma. All the material used as source of information has been acknowledged in the text.

Signature

To Sarah, Adiswa and Ivanna.

ACKNOWLEDGMENTS

I would like to express my gratitude to my supervisors Prof M.O Adigun and Prof S.S Xulu for their guidance, support and patience throughout the course of this work. I would also like to acknowledge the contribution of Gavin Rens through the discussion we would have on the sidelines of the FAIR meetings. To my fellow researchers in the centre for mobile e-Services for development, J. S. Iyilade, P. Mudali, M. B. Mutanga, P Tarwireyi, E. O. Olatunji, and A. O. Akingbesote I say, thank you very much for the support you gave me through out the development of the ideas that resulted in this thesis. To Tosin thank you very much all the help you gave in processing all the paper work required for registration and otherwise.

I also would want to acknowledge the support I got from my work colleagues at UKZN for all the support and encouragement they gave me. Special thanks to Anban Pillay and Serestina Viriri for being there when I needed advise on balancing my studies, work and family responsibilities

Last but not least I would like to express my gratitude to my wife S. T. Jembere for her understanding and being there for me through out the course of my studies.

TABLE OF CONTENTS

Page

Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

SYMBOLS

$\mathcal{L}$	Formal Language
K	Belief Set/Belief Base
$[K]$	Belief State
$a - world$	a set of worlds where the sentence, a, is true
$\mathcal{W}$	a set of possible worlds
$\mathcal{A}$	an algebra
A	a logical sentence
$\models$	logical entailment
κ	a kappa/ranking function
B^s	Bayesian Network Structure
B^p	a set of probability distribution correspobing to a Bayesian Network Structure
X_i	a variable/node in a Bayesian Network
π_i	a set of parent variables to variable X_i in a Bayesian Network
α_{ij}	an edge from X_j to X_i, $j \neq i$
D	a set of observation
$P^{\triangleleft}$	Unified Belief Change operator

ABBREVIATIONS

BLOG	Bayesian Logic
BN	Bayesian Network
DAG	Directed Acyclic Graph
DL	Descrption Logic
FOL	First Order Logic
FOPL	First Order Probabilistitc Logic
ILP	Inductive Logic Programming
KB	Knowledge Base
KR	Knowledge Representation
MEBN	Multi-Entity Bayesian Network
MLN	Markov Logic Networks
PGM	Probabilistic Graphical Model
PRM	Probabilistic Relational Models
RMN	Relational Markov Networks
SRL	Statistical Relational Learning
UBCOBaN	Unified Belief Change Operator for Bayesian Networks

ABSTRACT

Theory refinement in Probabilistic Knowledge Representations is the task of updating the Graphical Network structure in light of observations inconsistent with the current network structure. However, in the literature on Belief Change in Probabilistic Knowledge Representations, theory refinement is only thought of as a change in the model parameters when data consistent with the Network structure is observed. Such Belief Change is not rich enough to capture the semantics of Belief Change in dynamic domains. In dynamic domains, the actual network structure at any given time is unknown and is unobservable. Only the data emitted from the domain is observable. Further to the foregoing, the Belief Change Model needs to cater to both changes necessititated by the correction of incorrect Beliefs (Belief Revision) and changes necessitated by changes in the domain (Belief Update). This thesis hypothesised a Belief Change Meta-Model for Bayesian Network (BN) based Knowledge Representation in dynamic domains, and subsequently used the meta-model to define a Unified Belief Change Model for Bayesian Networks that caters for both Belief Revision and Belief Update of the Bayesian Network Structure.

The Belief Change Model was conceptualised by first modelling the evolving Bayesian Network structure as a dynamical system whose impetus for change is driven by the occurrence of some events in the domain. The derived Unified Belief Change Model was formally validated by analogy using the Qualitative Belief Change Model for dynamic environments and the theory of Partially Observable Markov Decision Processes (POMDP). It was also proven that the proposed Belief Change model meets the postulates for revision of *p-functions*.

Apart from arguing the efficacy of the proposed Unified Belief Change Model from a theoretical standpoint, this thesis also provides empirical evidence for the same. A Belief Change operator, the Unified Belief Change Operator for Bayesian Networks (*UBCOBaN*), based on the proposed Belief Change Model was developed. The operator was then used to illustrate how the model achieves Belief Change using a synthetic example with one (1) iteration of Belief Change. Further to the fore-going the operator was implemented in java and was used for evaluating the efficacy of the model in both Propositional Bayesian Networks and in Multi-Entity Bayesian Networks (MEBN). MEBN is a variant of First Order Probabilistic Logic (FOPL) this research chose to use for evaluating the proposed model for Belief Change in First-Order Probabilistic Knowledge Representations. The benchmark propositional Bayesian Networks used in the study were the *ASIA*, *ALARM*, *HAILFINDER*, *HEPAR II*, and the *ANDES* Bayesian Networks. The benchmark relational datasets considered for MEBN were the *CORA*, *WebKP*, *UW_std*, and *Financial_std* datasets. The results obtained showed that the proposed model adheres to the principle of minimal change (principle information economy) better than the classical Search-and-Score algorithm in all the afore-mentioned propositional Bayesian Networks and all the datasets considered for MEBN. The model was also found to be at least as agile as the classical Search-and-Score algorithm in instances where data inconsistent with the assumed network structure was observed. This was observed for all the benchmark propositional Bayesian Networks used in ths study, and all the relational datasets considered for MEBN. The results obtained for an investigation on whether Belief Update improves rationality of the proposed Unified Belief Change Model on propositional Bayesian Networks showed that the Unified Belief Change Model with Belief Update has superior performance compared to the one without Belief Update. However, the superior performance was not statistically significant at 95% Confidence Interval.

1. INTRODUCTION

"Knowledge has to be improved, challenged, and increased constantly, or it vanishes.,"
Peter F. Drucker.

1.1 Introductory Background

Knowledge acquisition, representation and management for supporting today's increasingly open and dynamic computing environments pose a lot of challenges to knowledge Engineers. The challenges emanate from the fact that open and dynamic computing environments are dogged by uncertainty, vagueness, complexity, and massive data. These factors render manual revision and updating of the Knowledge Bases infeasible. This has resulted in Artificial Intelligence researchers seeking solutions for enabling open and dynamic computing environments to automatically update and revise their Knowledge Bases. The emergence of such solutions will not only be beneficial to open and dynamic computing environments, but it will also contribute towards extending the frontiers of knowledge in the field of Artificial Intelligence towards Artificial General Intelligence (AGI)(Goertzel, 2014). One of the major impediments to the development of Artificial Intelligence has been their over dependence on human beings for the knowledge needed by intelligent systems. This has led to the problem dubbed the Knowledge Acquisition Bottleneck (Možina et al., 2008). In principle, Artificial Intelligence seeks to achieve systems that have the hope of overcoming their problems without human agents holding their hands constantly. "If intelligence is to be engineered, this is simply a requirement" (Korb & Nicholson, 2004).

The dependence on human agents as the only source of knowledge was one of the major challenges with the first generation expert systems. It made them too brittle. When the problem domain had changed or the problem focus had changed to include anything new, the system would simply break. These systems lacked the ability to learn, partly because of their limited data handling capabilities. The need for intelligent systems that can learn has now been brought into sharper focus by the availability of vast amounts of data and computational power that can be used to enable systems to revise and update their Knowledge Bases and adapt to the context of computing. The challenge now is how intelligent systems can be enabled to rationally revise and update their Knowledge Bases using these vast amounts of data, in an environment characterised by uncertainty, vagueness,and complexity. This problem cannot only be viewed as a problem of the nature and source of knowledge - an epistemology problem, but it is also partly an ontology problem. Hence, though this thesis seeks to address an epistemology problem, it starts by arguing the need for a knowledge representation that inherently handles uncertainty, ambiguity and complexity, and supports revision and update of beliefs held in Knowledge Bases. This thesis argues that even though probability and its related concepts, such as possibility (Dubois & Prade, 2014), and plausibility (Collins & Michalski, 1989) are not ontological, having a knowledge representation that can seamlessly integrate these concepts into a representation of the ontological aspects of a domain, provide an epistemological convenience that enables systems to automatically revise and update their knowledge about the domain just as human beings do. Humans revise and update their beliefs about the world through observation of the many uncertain, incomplete, vague and complex aspects of the world.

Underlying the human agent's ability to deal with uncertainty, inconsistencies, incompleteness, and complexity is their ability to implicitly assign some degree of belief to what they observe and the logical consequences of the observations. This is done in the context of the beliefs about the world that the agent holds prior to observations. There are a lot of frameworks to Belief Change that have been discussed in literature over the years, but this thesis, in its bid to address the epistemology problem in open and dynamic computing environments advocates for a framework based on probabilistic logic and a Bayesian approach to Belief Change. The choice of a Bayesian approach is not withstanding all the arguments against it, which will be duly discussed in later chapters.

1.2 Probabilistic Logic Representation

While Probabilistic models inherently deal with uncertainty, inconsistencies and complexity in many real world domains, they mainly operate at propositional level. This means that probabilistic models are not rich enough to capture relationships between classes of objects. First Order Logic on the other hand is highly expressive but struggles with handling uncertainty, inconsistencies and complexities. It has become apparent that it is desirable that these logical systems should be integrated to create a logical framework that is expressive enough for real world domains. Over the past few years, several First Order Probabilistic Logic (FOPL) frameworks have been proposed. A treatise of these frameworks will be given in Chapter 2. The discussion in this chapter is limited to what is relevant here to position this thesis in the maze of existing literature on Belief Change and probabilistic logic. The position taken here is to view an FOPL as an extension of Probabilistic Graphical Models with First Order Logic (FOL) semantics. Works that have emanated from the field of Statistical Relational Learning (SRL) have over the years proposed a few Probabilistic Logic frameworks such as Probabilistic Relational Models (PRMs) (Getoor, 2000), Markov Logic Networks (MLN) (Richardson & Domingos, 2006), Multi Entity Bayesian Networks (MEBN) (Laskey, 2008), etc. However, the efforts in these works concentrated on the definition of representation semantics for the frameworks and not much has been directed towards enabling Belief Change in such representations.

1.3 The Belief Change Problem

The Belief Change problem became of interest in Philosophical Logic and Artificial Intelligence in the middle of the 1980s. The focus of the research has been mainly to understand how an agent should change its beliefs as a result of getting new information. The key principle underlying Belief Change is that the agent should make minimal changes to its beliefs in the face of new information. That is, the agent should not give up its beliefs unnecessarily unless there is compelling evidence to do so. This is popularly known as the principle of minimal change or information economy. An agent that adheres to this principle is therefore said to be rational

Belief Revision and Belief Update are two of the most studied models of belief change. In principle both belief change types try to explain the source of incorrect beliefs at any given time, but make different assumptions about the sources of incorrect beliefs (Boutilier, 1998). If beliefs about the world are simply incomplete or mistaken, steps must be taken to correct the misconception. A process of rationally correcting such is what is known as Belief Revision and the AGM theory (Grdenfors & Makinson, 1988) is the best-known characterisation of such in Qualitative Belief Change. On the other hand, if beliefs about the world were once correct and complete, but are now incorrect or incomplete owing to some changes in the world, steps must be taken to update the beliefs to reflect the current state of the world. Katsuno and Mendelzon (1991) proposed a general characterisation of Belief Update that provides the constraints that must be satisfied to reflect the changes in the domain.

Both Belief Revision and Belief Update have received a lot of attention in literature. However, Knowledge Representation for Open and Dynamic Computing Environments require mechanisms for handling both Belief Revision and Belief Update, which have not received much attention in literature. Given the foregoing, this thesis aims to address the Belief Change problem in open and dynamic computing environments that this research coined the Knowledge Lag problem. The knowledge lag is defined as the gap that exists between the knowledge held in a given knowledge base and that which is the true state of affairs in the world. The knowledge lag emanates from the fact that in an open and dynamic computing environment services, new concepts, relationships, and practices can be added at will. As a result, if there are no mechanisms to automatically update and/or revise knowledge bases, the gap between the knowledge required to enable autonomous systems to perform satisfactorily and the knowledge represented in knowledge bases will keep widening. However, as the knowledge is being automatically evolved, a lot of ambiguities and uncertainties arise due to the openness of the environment and the resultant knowledge bases are bound to become more complex. Another sub-problem to the knowledge lag problem arises as the gap between the knowledge captured in the knowledge bases and the knowledge that is actually useable in the presence of increasing complexity keeps widening. This thesis coined this problem the Knowledge Gap problem.

Over the years Probability Theory has emerged as a sound mechanism for handling ambiguity, uncertainty and complexity in information systems. This has seen techniques that have emanated from the field of Mathematical Statistics being used, both as a mechanism for enabling computers to glean knowledge from huge amounts of data and as a mechanism for enabling machines to handle uncertainty and complexities in knowledge reasoning. Classical knowledge representation languages do not handle uncertainty and deductive approaches to knowledge inference are known to break down under increasing complexity. To address t his problem it is widely accepted that a framework with a principled way of handling uncertainty will do. Probabilistic Graphical Models (PGMs), such as Markov and Bayesian Networks, have been studied as a means for representing knowledge in dynamic environments. The major weakness with PGMs is that they represent knowledge at a propositional level and hence like propositional logic they lack flexibility. This calls for a first order version of PGMs, since First Order Logic is known to be flexible. In the quest for producing knowledge representation languages that are First Order and can handle uncertainty in a principled way, efforts have been both on enabling FOL to handle uncertainty and enriching PGMs to handle First Order Logic. These efforts are resulting in frameworks that are more or less the same. Learning algorithms from PGMs are adapted to address the Belief Change problem in these languages. However, no effort has explicitly been put into investigating whether these learning algorithms conform to Belief Change principles and no framework has been proposed on formalising the knowledge evolution problem in these knowledge representations. The work presented in this thesis is an effort to provide an iterative Belief Change Model for evolving Probabilistic Knowledge Representations(Knowledge based on PGMs) that conforms to the principles of rational Belief Change.

The remainder of this introduction provides the thinking that produced this thesis, and the space this thesis fits in, in the existing philosophical views to logic, epistemology, and Belief Change.

1.4 Research Questions

In addressing the Belief Change problem in Probabilistic Knowledge Representations for Open and Dynamic Computing environments, this research sought to answer the following research question:

How can a Unified Belief Change Model for Probabilistic Knowledge Representations that caters for both Belief Revision and Belief Update be designed?

In order to comprehensively answer the fore-going research question, the following sub-research questions were investigated:

1. How can Belief Change principles that have emanated from classical Belief Change be used to model Belief Change in Probabilistic Knowledge Representations?

2. Does a solution based on classical Belief Change principles result in more *rational Belief Change* (adheres to the principle of minimal change) compared to classical Bayesian based Probabilistic Graphical Model Structure learning algorithms for both Propositional Bayesian Networks and First Order Probabilistic Knowledge Representations?

3. Is Belief Update important for Belief Change in Probabilistic Knowledge Representations?

1.5 Research Aim and Objectives

Given the fore-going discussion, this work aimed to develop an iterative Belief Change model for evolving Knowledge Representations in open and dynamic computing environments.

The specific objectives were to:

1. do a literature survey to establish the state of the art in knowledge representation and Belief Change in open and dynamic computing environments,
2. develop a Unified Belief Change model that caters for both Belief Update and Belief Revision in Open and Dynamic computing environments,
3. argue the efficacy of the developed model from its theoretical underpinnings,
4. develop and implement a Belief Change operator based on the developed Unified Belief Change Model,
5. empirically evaluate the Belief Change Model using Benchmark datasets.

1.6 Positioning of this thesis

Some of the hotly debated issues in Belief Change are: how epistemic states should be modelled, what the objects of belief should be, and what is the nature of the relata of the degree of belief relation. It is common in Belief Change literature to assume that a belief is a relation between an epistemic agent at a particular time to an object of belief (Huber, 2009). The discourse in this section will start by giving this thesis position on what is the nature of the relata of the degree of belief.

The degree of belief is a relation between an agent, the objects of belief and a numerical value at a given time. The numerical value serves to give the strength with which the agent(s) believe the truthfulness of various propositions. The higher the agent's degree of belief in a particular proposition the higher the confidence in the truth of the proposition. However, the precise meaning of a given degree of belief depends on the underlying theory of degrees of belief. For instance, if probabilities are used to capture the degrees of belief, a probability of 0.5 that a coin comes up heads may in some theories indicate ignorance, and indicate that the coin is fair in others. This thesis' position on the degree of belief is based on the theory of subjective probabilities. This theory, advocates that the degrees of belief should satisfy the laws of probability. Although this position has no principled way of representing ignorance, it suffices in solving the Belief Change problem that is addressed in this thesis.

This thesis takes propositions as the objects of belief. Thus, beliefs are defined over propositions regardless of the language used to model the domain. We define a proposition to be a set of possible worlds or truth conditions (Huber, 2009). This thesis holds the position that in any world there exists a non-empty set of possibilities, $\mathcal{W}$, such that at any given point in time there is exactly one element of $\mathcal{W}$ that corresponds to the actual world. However, this element is not observable and as a result a belief state is kept as a subjective probability distribution over all the possible worlds. However, this belief state does not sufficiently define the epistemic state of the world. Since the Belief Change Model defined in this thesis is for dynamic environments, the belief state needs to model the propensities of the belief dynamics in the epistemic state. This thesis postulates that these propensities can be captured through event plausibilities relative to the hypothesised state of the world. This is one of this thesis' major deviations from classical Belief Change theory. Although this hypothesis has been studied in Qualitative Belief Change, this thesis is the first work towards studying this phenomenon for Quantitative Belief Change and adapting it to solve the Belief Change in Probabilistic Graphical Models based Knowledge Representations.

1.7 Other Perspectives

The best developed account of the degrees of belief is the theory of degrees of subjective probabilities. In this theory, Bayesian Conditionalisation (Greaves & Wallace, 2006; P. M. Williams, 1980) is taken as the means for effecting Belief Change. One of the key weaknesses of conditionalisation is that it is not feasible when the new information is observed with uncertainty. This, coupled with the fact that subjective probabilities have no principled way for representing ignorance, lead to proposals for other measures for degrees of belief. This takes the discussion to Dempster-Shafer (DS) function (Dempster, 1968; Shafer, 1976) and Possibility Theory (Dubois and Prade, 1988). In DS function an agent's beliefs about a proposition are divided into three (3) mutually and jointly exclusive parts; (i) a part that favours A, (ii) a part that favours $W\backslash A$, and (iii) a part that neither favours A nor $W\backslash A$ (denoted by I). $P(A)$ quantifies the degree of belief in A and $P(W\backslash A)$ quantifies the degree of belief in $W\backslash A$. Thus, $P(I) = 1 - P(W) - P(W\backslash A)$. This means subjective probabilities can be seen as DS belief function without ignorance.

Possibility theory postulates that a proposition is at least as possible as all of the possibilities it comprises and no more possible than the most possible possibility. To a large extent it is comparable to probability theory. However unlike probability theory, it uses a pair of dual set functions, *possibility* and *necessity* measures instead of only one. Intuitively, possibility theory relates more to humans' perception of the degree of feasibility or ease of attainment (Zadeh, 1977) rather than how probable an event is. On the other hand, probability is associated with likelihoods of events and degrees of belief. It is on the backdrop of the fore-going that this thesis chose to use probability over possibility as a measure of the degree of belief.

Other perspectives that compete with quantitative degrees of belief are qualitative degrees of belief. Qualitative degrees of belief are often represented as ranking functions (Spohn, 2012). Ranking functions partition the set of possible worlds into sets of possibilities that are mutually exclusive and jointly exhaustive. The sets are then ordered with respect to their plausibilities. The first set contains the worlds that are considered the most reasonable candidates for the actual worlds and this set is said to be containing the possibilities that cannot be disbelieved.

1.8 Contributions of the Study

The hype about the Fourth Industrial Revolution is pushing researchers to begin to ask some questions about how some of the day-to-day tasks which were traditionally reserved for human beings can be done by machines. One such question that this thesis is based upon is: Can machines do science? If they can, what are the necessary and sufficient preconditions to make that possible? This thesis postulated two (2) necessary preconditions: (i) there should be a knowledge representation that can inherently handle uncertainty, and ambiguity inherent data, (ii) the scientific process for knowledge discovery should adhere to the principle of minimal change.

In arguing the afore-mentioned notions, this thesis makes the following contributions towards extending frontiers of knowledge in the field of automatic evolution of Knowledge Representations:

First, the thesis took a position that First Order Probabilistic Logic (FOPL) is the ideal knowledge representation framework for enabling machines to automatically refine their knowledge, and then formalised the logical bonds between Belief Change in classical logic and Belief Change in FOPL that uses Bayesian Networks as their underlying knowledge representation framework. The possible worlds view to propositions provided a natural glue between the two worlds.

Second, the thesis developed a Unified Belief Change Model for automatic evolution of probabilistic Knowledge representations that caters for *Iterative* Belief Revision and Belief Update. Bayesian Conditionalisation was used to achieve Belief Revision, and event semantics were used to achieve Belief Update. To ensure iterative Belief Change the model defined an *Epistemic Space* that enabled it to return an *Epistemic State* rather than a Belief Set or Belief Base as the output of a Belief

Change process. The Unified Belief Change Operator for Bayesian Networks (UB-COBaN) parenciteJembere2016 was developed based on the proposed Belief Change model. Experimental results obtained for both Propositional Bayesian Networks and Multi-Entity Bayesian Networks (an instance of FOPL used for experimentation in this study) showed the importance of using the Epistemic States in the Belief Change process in ensuring Minimal Change and faithfulness to the processes emitting the data used in the Belief Change process.

1.9 Overview of the Chapters

This thesis is organised as follows: The next two chapters cover the foundational concepts prevailing in this thesis. Chapter 2 presents the conceptual basis for knowledge representation over which the proposed Belief Change model is defined. It starts by establishing the need for Knowledge representation that inherently handles uncertainty. Different First Order Probabilistic Knowledge representations are discussed. Chapter 3 provides a treatise of the findings from a survey on the literature on Belief Change in dynamic domains. This chapter sets the platform for the introduction of Belief Change to First Order Probabilistic Knowledge representations. The chapter also discusses other alternative views to Belief Change in dynamic domains.

Chapter 4 presents the Belief Change Model for First Order Probabilistic knowledge representations that was developed in this study. The Chapter discusses the logical bonds between classical logic and Bayesian Networks and then latches onto this platform to define a Belief Change model for Bayesian Network-based knowledge representations. Chapter 5 presents a Belief Change operator based on the model defined in Chapter 4. The Operator takes advantage of the progress that has been made in the literature on Bayesian Structure Learning. An illustration of how the operator works using a toy example is also presented in Chapter 5.

Chapter 6 presents the empirical results on the evaluation of the proposed Belief Change Model on Benchmark Propositional Bayesian Networks. The evaluation is meant to establish that the operator adheres to the principle of minimal change and is agile enough to adapt to change if there is a change in the underlying process emitting the data. The evaluation is benchmarked on the classical search and score algorithm implemented in Banjo [1] Chapter 7 evaluates the proposed model in Multi-Entity Bayesian Networks, a Bayesian Network based First Order Probabilistic Knowledge Representation. The chapter discusses the implementation of the operator for MEBN and evaluates its adherence to the principle of minimal change and agility to structure changes when the underlying process emitting the data changes.

Chapter 8 summarises the thesis, draws some conclusions from the study conducted in this thesis and gives some future research directions.

[1]https://users.cs.duke.edu/ amink/software/banjo/

2. KNOWLEDGE REPRESENTATION IN OPEN AND DYNAMIC ENVIRONMENTS

"Knowledge is an unending adventure at the edge of uncertainty"
Jacob Bronowsk

2.1 Introduction

The pervasiveness of uncertainty, incompleteness and complexity in net-centric computing environments is pushing the envelope on how the knowledge needed to make such systems autonomous should be represented. Over the years a lot of work has been directed at the development of languages for knowledge representation in net-centric environments. This has resulted in the concept of ontologies being adopted in Computer Science. In Computer Science, an ontology is defined as the conceptualisation of a domain. Quite a lot of ontology languages have been defined over the years, but the Web Ontology Language (OWL)(Antoniou & van Harmelen, 2004) has emerged to be the most widely used. The underlying knowledge representation used in OWL is Description Logics (DL), a decidable variant of First Order Logic (FOL). First Order Logic is believed to be expressive enough to represent all forms of knowledge needed by intelligent systems, but FOL logic is known to be very brittle under uncertainty, ambiguity and increasing complexity. First Order logic is restricted to representing facts that are absolutely true (Getoor, Friedman, et al., 2001). On the other hand, probabilistic models are known to be a very robust mechanism for handling uncertainty in decision making systems, but they inherently lack the expressiveness to handle First Order semantics.

Over the past two decades probabilistic graphical models have become increasingly popular, both as a language for representing knowledge about uncertain phenomena and architecture for efficient inference algorithms. The turn of the twenty first century saw a lot of efforts to combine FOL, and Probabilistic Graphical Models to take advantage of the expressiveness of FOL and the inherent ability to handle uncertainty, ambiguity, and complexity in Probabilistic Graphical Models. This has resulted in a new field of Artificial Intelligence (AI) that has been dubbed Statistical Relational Learning (SRL). The Knowledge representation techniques that have emanated from the field of SRL either extends FOL with PGM semantics or extend PGMs with FOL semantics.

This chapter discusses the conceptual basis for the knowledge representation that serves as the platform over which Belief Change in Open and Dynamic computing environments will be defined. The discussion in this chapter seeks to argue that SRL-based knowledge representations are ideal for knowledge representation in Open and Dynamic computing due to the fact they have the expressiveness of FOL and can inherently handle uncertainty, ambiguity, and increasing complexity, which are pervasive in any real-world domain. The argument will be presented from logical and intuitive perspectives. This chapter will also endeavour to provide conceptual clarity on the ontological aspects of probabilistic knowledge representations and their separation from the epistemological aspects of the representations which are often integrated with the representation. This may appear to be conceptual clumsiness from a philosophy of science perspective, given that probabilities (degrees of belief) are widely not taken as ontological (Nau, 2001; Rosinger, 2010). Probabilities are thought to be just an epistemological convenience that science uses to revise and update theories (beliefs about the world). However, the strength of First Order Probabilistic Knowledge representation is drawn from such philosophical clumsiness, and it is in the hope of providing clarity in such clumsiness that this chapter is presented. Several SRL languages will be discussed with the aim of identifying the common ingredients that can be used to give a general taxonomy of the SRL languages based on the underlying knowledge representation approach. The discussion will also explore how these representations render themselves to handling knowledge in open and dynamic environments and rational Belief Update and Belief Revision. This discussion will start with a characterisation of Open and Dynamic Computing Environments and then digress into discussing how knowledge should be represented to ensure rational Belief Change in knowledge bases for Open and Dynamic Computing Environments.

2.2 Open and Dynamic Computing Environments

This section characterises what this thesis refers to as Open and Dynamic Computing Environments (ODCE). An open computing environment is characterised as an environment where documents, systems and services may appear, change or disappear at any time and thus no assumption can be made about the content protocol, or even availability or existence of entities in the environment (Palmisano et al., 2008). Further to the above, open computing environments have no boundaries between legitimate and illegal users of a system. Owing to the openness of the environment, the notion of a common knowledge representation catering for the diverse range of entities in the domain becomes untenable and thus necessitates the intelligent adaptation of the knowledge representations to the status quo of the domain. Thus, the system also becomes dynamic since the configuration of these systems is constantly changing.

A survey of related literature on Open and Dynamic Computing Environments found the following challenges towards the goal of supporting knowledge-based decision-making in Open and Dynamic computing environments (Jembere, Xulu, & Adigun, 2010): (i) *Certainty of uncertain, ambiguous and inconsistent data*, (ii) *The dynamic nature and complexity of the computing environment*, (iii) *Impracticality of manual management of Belief Change in Knowledge Bases.*

2.2.1 Certainty of uncertain, ambiguous and inconsistent data

Much of the information in Open and Dynamic computing environments is uncertain, ambiguous, often incorrect or only partially correct raising issues related to trust and credibility of inferences drawn from such information (Laskey, 2008). Using deterministic ontologies to represent such knowledge leaves a lot to be desired. Uncertainty representation and reasoning in such environments have the promise of discounting the effect of these data imperfections and provide a proof theory over knowledge bases in such environments.

2.2.2 The dynamic nature and complexity of the computing environment

Knowledge representations in Open and Dynamic environments need mechanisms for incorporating new knowledge into the KBs. This is due to the following two reasons: (i) owing to the complexity of the environment and incredibility of the knowledge sources, the system's beliefs about the world may simply be mistaken or incomplete, (ii) the system's beliefs about the environment might have been correct at some point in time, but the belief may have become inaccurate due to changes in the world, rendering certain facts true and falsifying some. (Laskey, 2008). Further to this, in dynamic environments, the truthfulness of the system's beliefs might be situational, which will require situational knowledge representation and reasoning (e.g. situational reasoning (Laskey, 2008), and Defaults (Lukasiewicz, 2002)

2.2.3 Impracticality of manual management of Belief Change in Knowledge Bases

The complexity of the environment and the size of the Knowledge Bases in open computing environments make manual update and revision of Knowledge Bases impossible owing to the following reasons: (i) Knowledge representation in general, requires that the person effecting the changes to the knowledge base be both a knowledge engineer and a domain expert and very few people can be both; (ii) due to the collaborative nature of the environment manual belief changes by different knowledge engineers is likely to result in different KBs. This is because knowledge engineers have different views on how a certain change should be implemented resulting from differences in background knowledge, personal preferences, subjective opinions, etc,(Flouris, Plexousakis, & Antoniou, 2006); (iii) Another source of problems for manual revision and update of knowledge bases is the complexity of modern-day Knowledge representations: These knowledge bases are usually developed by several knowledge engineers or even teams comprising different expertise; (iv) in highly dynamic computing environments, ontology changes are so frequent that by the time the engineers finish effecting the change the updated KB may still be lagging behind the current state of knowledge in the domain.

The above discussed characterisation of Open and Dynamic computing environments calls for a new thinking on how knowledge representation for such environments should be done. It is in view of the above discussion that this thesis argues for the following desiderata for knowledge representation in open and dynamic computing environments:

i. Knowledge Representation should be able to represent knowledge about entities that are related to each other, and reason about the knowledge in the presence of uncertainty, incompleteness, ambiguity and complexity.

ii. The knowledge representation framework should support situational reasoning.This is meant to counter the effect of complexity of the knowledge representation, and

iii. Knowledge representations should have the capability of being easily integrated with a rational and objective operator for automatic iterative Belief Change.

The discussion on the above will be revisited later when choosing a Knowledge representation framework for this study.

2.3 Inconsistencies, Uncertainty, Ambiguity and Knowledge Representations

Uncertainty, ambiguity, inconsistency, and complexity, rather than being exceptions, are typical default characteristics of Open and Dynamic computing environments. Classical knowledge representation approaches do not cater to these characteristics. This calls for the development of mechanisms that enable next generation computing environments to account for these characteristics in representing and reasoning upon the knowledge in their Knowledge Bases.

In classical logic inconsistencies are taken as bugs or defects in the knowledge base. Traditional ways of dealing with inconsistencies involve having to remove some information from the knowledge base. At times, approximate reasoning is used but it often forfeits correctness. Probabilistic approaches to knowledge representation addresses the consistency problem by relaxing the proposition in the knowledge base if the world violates any of the propositions. That is, violated propositions are only made less likely but not impossible.

Probability has emerged as the natural candidate to represent uncertain phenomena. Owing to its promises towards this goal, a lot of research efforts have been directed towards introduction of probabilities in knowledge representations though hindered by scepticism on the ontological aspect of probability (Roelofs, 1935) and tractability of inferences and feasibility of representation (da Costa et al., 2005). Progress has been made on this frontier, but a philosophical discussion on the relata between logic and probability is still a hotly debated topic.

2.4 Logic and Probability

Although the formal behavior and the calculus of probability is for the most part uncontroversial, their interpretation has largely been controversial. Probabilities can be interpreted as either (i) statistical statements or (ii) degrees of belief. As proposed in (Roelofs, 1935), this thesis uses the term Statistical Probability to refer to the former and Propositional Probability to refer to the latter. These two notions of probability are quite distinct. Statistical probabilities are defined over sets of individuals and do not relate to particular individuals. They are a reflection of the statistical regularities in the world. An example would be, 80% of the papers referenced by papers on reinforcement learning are in the category of reinforcement learning. This is a statistical assertion about the proportion of papers that are cited by papers on reinforcement Learning. Propositional probabilities, on the other hand, are attached to propositions about particular individuals. An example would be: the probability that a paper with 80% of its citation from reinforcement Learning is on Reinforcement Learning is 60%. This is an assertion about the degree of belief, and its truthfulness determined by the subjective state of the agent making the statement. There is no connection between an agent's subjective belief and the objective state of the world.

From a philosophical point of view, the difference between these concepts should be amplified to provide clarity in order to avoid theoretical clumsiness. This has over the years led proponents of probability in AI choosing to side-step the empirical foundation of propositional probabilities. The belief is the only constraint that should be applied to subjective probabilities is obeying the axioms of probability. This view ignores the issue of the source of these probabilities and how they relate to the objective state of the world which pertains to empirical experience. This thesis submits that for rational Belief Change in Knowledge Representations in Open and Dynamic Computing environments, there is a need for integrating the interpretations of probability into forming a model for rational Belief Change in Open and Dynamic computing environments.

In the literature on philosophy, there is already a well developed formalism for integrating logic and probability. The basic claim of this formalism is that for any two propositions A and B there is an objective, logical relation of *partial* entailment between A and B measured by a unique conditional probability $P(A|B)$. The key known proponents of this formalism are Canarp (Carnap, 1950) and Keynes (Keynes, 1921), but the position is referred to as the Carnapean position in philosophy literature (Bradley, 2018). This position as presented by carnap uses some version of the principle of indifference to logically infer the conditional probability values. This approach to determining probabilities is rejected in most contemporary literature on First Order Probabilistic logic. Contemporary approaches to Statistical Relational Learning advocate that the probabilities should be defined by users or they should come from data. This thesis submits that these probabilities should come from data. Although contemporary techniques for integrating probability and logic reject Carnap's approach to determining probabilities, it turned out that his approach of using *possible worlds semantics* to interpret both logic and probability provides the glue that relates logic and probability.

In classical logic, to interpret terms and formulae of a FOL language $\mathcal{L}$ it is neccessry to consider $\mathcal{L}$structures, which are also known as *possible worlds.* For instance $flies(tweety)$ is true in a given possible world, $\mathcal{W}$, if the individual denoted by $tweety$ belongs to a set of things that fly. Thus, the set of things that fly defines the possible worlds for things that fly. Given such semantics, the probability that an individual represented by the constant tweety flies is defined as P(flies(tweety)). In the objective world its either the individual tweety flies or it does not. However, from the empericist position, we can use counting processes on the possible worlds to get some degree of belief on whether the individual,tweety, flies or not, by counting the number of worlds where tweety flies. From a statistical point of view, the probability of an individual, *tweety1* of type tweety, can be estimated by counting the number of possible worlds in which individuals of type tweety have been observed flying.

Rational Belief Change in First Order Probabilistic Models is the focus of this thesis. However, in order to come up with some algorithms for such, it is important for one to understand how beliefs are represented in the Probabilistic Logic fomalisms.

2.5 Why First Order Probabilistic Logic

Logic and probability theory are two of the main tools in the formal study of reasoning (Demey, Kooi, & Sack, 2016). Logic is a schema for defining languages for describing and reasoning about entities in a domain. It offers a qualitative (structural) perspective on representation of the knowledge and inferences concerned with absolutely certain truths and inferences about the domain. Thus, classical logic has no apparatus to handle uncertainty and inconsistencies that are pervasive in real world domains. The most widely used logical system is First Order Logic. First-order logic is applied by defining a set of axioms, or sentences that make assertions about a domain. The axioms, together with the set of logical consequences of the axioms, comprise a theory of the domain. Until referents for the symbols are specified, a theory is a syntactic structure devoid of meaning. An interpretation for a theory specifies a definition of each constant, predicate, and function symbol in terms of the domain. Each constant symbol denotes a specific entity, each predicate denotes a set containing the entities for which the predicate holds, and each function symbol denotes a function defined on the domain. The logical consequences of a set of axioms consists of the sentences that are true in all interpretations, also called the valid sentences.

First Order Logic (FOL) is known to have the ability to represent entities of different types interacting with each other in varied ways. A first-order theory enforces truth-values for the valid sentences and their negations, but offers no means of evaluating the plausibility of other sentences that may not necessarily be true but are probably true. Plausible reasoning is fundamental to intelligence owing to the fact that uncertainty is more of a rule than an exception in the real world. FOL lacks a theoretically principled way of reasoning under uncertainty.

Probability theory naturally has the apparatus to deal with uncertainty in real world domains. As a result graphical probability models have become popular as a parsimonious language for representing knowledge about uncertain phenomena, a formalism for representing probabilistic knowledge in a logically coherent manner, and an architecture to support efficient algorithms for inference, search, optimisation, and learning. A graphical probability model expresses a probability distribution over a collection of related hypotheses as a graph and a collection of local probability distributions. The graph encodes dependencies among the hypotheses. The local probability distributions specify numerical probability information. Together, the graph and the local distributions specify a joint distribution that respects the conditional independence assertions encoded in the graph, and has marginal distributions consistent with the local distributions.

Probabilistic Graphical Models by themselves are probabilistic extensions of propositional logic. As a result, like propositional logic, they are limited in that they cannot represent objects and relations between them, and that is certainly an important part of rationality. They are insufficiently expressive to reason about varying numbers of related entities of different types, where the numbers, types, and relationships among entities usually cannot be specified in advance and may have uncertainty in their own definitions. This makes probability theory an incomplete theory of rationality. This lack of expressiveness limits application of Probability Graphical Models in representing knowledge about real world domains. On the other hand, although First Order Logic is very expressive, it in itself lacks ability to reason about evidence. Probability in the form of Bayesianism is a theory of evidence. The big question is: Can they be combined to get a complete theory of rationality? Against this background a number of languages that integrate Bayesian Networks with First Order Logic have been proposed. Although there are Philosophical arguments against integrating the two, practical applications have shown that integrating logic and probability works for many real word applications. This has prompted a lot of studies directed towards combining First Order Logic and Probabilistic Graphical Models.

2.6 First Order Probabilistic Logic (FOPL)

First Order Probabilistic Logic solutions can be categorised along two (2) dimensions. The first dimension is based on the nature of the underlying formalism for knowledge representation, which could be either rule-based or frame-based formalisms. Rule-based formalisms emanated from efforts extending First Order Logic with probabilities. Hence, these tend to add probabilities as weights to FOL statements. Syntactically, rule-based models are indistinguishable from FOL, except that each formula will have a weight attached to it. Frame-based solutions emanated from efforts aimed at extending Probabilistic Graphical Models to handle FOL. Hence, they tend to focus of the objects and the relationships between them. They use Probabilistic Graphical Models to capture the dependence structure between attributes of the objects and the relationships between them. The second dimension along which FOPLs can be categorised is the nature of the underlying graphical structure used to capture the probabilistic aspect of the logic. The graphical models can either be directed or undirected. Directed models use Bayesian Networks as their underlying knowledge representation structure, whereas undireceted models use Markov Networks. Both Bayesian networks and Markov networks model the joint probability among random variables by decomposition. The goal is to simplify the joint probability distribution, as well as preserve interesting dependencies that can then be used to model the statistical regularities of the domain.

An analysis of existing solution along these dimensions produce the following categories of FOPL solutions

i. Rule-based Undirected Models

ii. Frame-based Undirected Models

iii. Rule-based Directed Models

iv. Frame-based Directed models

2.7 Undirected First Order Probabilistic Logic

The Undirected models are based on the Markov networks. These models model symmetric, non-causal interaction between attributes of an object and/or relationships between objects. To create the background needed to understand these models the next section will discuss the basic concepts of Markov Networks.

2.7.1 Markov Networks

A Markov network is a model for the joint distribution of a set of variables $X = (X_1, X_2, ..., X_n) \in X$ (Pearl, 2009). It is composed of an undirected graph G and a set of potential functions ϕ_k. The graph has a node for each variable, and the model has a potential function for each clique in the graph. A clique is a subgraph where every two vertices are connected to each other. Cliques enable the probability distribution to factorise into a probability distribution that is easier to parameterise. A potential function is a non-negative real-valued function of the state of the corresponding clique. The joint distribution represented by a Markov network is given by

$$P(X = x) = \frac{1}{Z} \prod_{k} \phi_k(x_{\{k\}}) \tag{2.1}$$

where $x_{\{k\}}$ is the state of the k^{th} clique. Z, known as the partition function, is given by $Z = \sum_{x \in X} \prod_k \phi_k(x_k)$. Markov networks are often conveniently represented as log-linear models, with each clique potential replaced by an exponentiated weighted sum of features of the state, leading to

$$P(X = x) = \frac{1}{Z} exp(\sum_{j} w_j f_j(x))) \tag{2.2}$$

A feature may be any real-valued function of the state. In the most direct translation from the potential-function form (Equation 2.1), there is one feature corresponding to each possible state $x_{\{k\}}$ of each clique, with its weight being $log \phi_k(x_{\{k\}})$.This representation is exponential in the size of the cliques. However, we are free to specify a much smaller number of features (e.g., logical functions of the state of the clique), allowing for a more compact representation than the potential-function form, particularly when large cliques are present.

Undirected Logic models make use of the concept of Markov Networks, hence no formula has its own probability explicitly stated. The probability of each possible world is defined in terms of its features, where each feature has an associated real value parameter.

Undirected logic models generally have one major advantage over directed models. Undirected models avoid the the difficulties of defining a coherent non-cyclical generative models for graph structures as required in directed graphical models. However, this comes at the price of more expensive learning operations.

2.7.2 Rule-Based Undirected Models

Rule-based Undirected Models extend FOL statement with Markov Networks. Markov Logic Networks (MLN)(Richardson & Domingos, 2006) are the only known example of such models. Owing to this fact the discussion in this section is going to focus on Markov Logic Networks. MLNs define a probability distribution over as set of worlds as follows (Richardson & Domingos, 2006):

An MLN is a set of pairs (F_i, w_i), where F_i is a formula in first-order logic and w_i is a real number. Together with a finite set of constants $C = \{c_1, c_2, c_3, ..., c_{|C|}\}$, defines a Markov Network $M_{\mathcal{L},C}$ as follows:

1 $M_{\mathcal{L},C}$ contains one binary node for each possible grounding of each predicate appearing in $\mathcal{L}$. The value of the node is 1 if the ground atom is true, and 0 otherwise

2 $M_{\mathcal{L},C}$ contains one feature for each possible grounding for each formula F_i in $\mathcal{L}$. The value of this feature is 1 if the ground formula is true and 0 otherwise. The weight of the feature is the w_i associated with F_i in $\mathcal{L}$.

Learning structure in MLNs, is done through Inductive Logic Programming (ILP)(Muggleton, 1991). In ILP, hypotheses are constructed through refinement operators that add and remove literals from clauses. Learning is done from data in relational databases by making the closed world assumption. Thus, if a ground atom is not in the database, it is assumed false. Belief Change in such representation will have to make use of inductive logic programming. Nothing much has been done in literature on how ILP relates to rational Belief Change. One effort that investigates this is the work by Pagnucco and Rajatratnam (2005) which investigated Inverse resolution as Belief Change operator.

2.7.3 Frame-based Undirected Models

Frame-based Undirected Models extend Markov Networks with FOL semantics. Instead of defining the FOL statements for a given domain, these models endeavour to come up with a joint probability distribution for the entire collection of related entities. The resulting model is Markov Network over a relational data set. The goal is to use the Markov Network to model the relational structure of the domain, and the same can easily model complex patterns over related entities. One example of a frame-based directed model is Relational Markov Networks(RMN) (Taskar et al., 2007).

An RMN specifies a conditional probability distribution over all of the labels of all entities in an instantiation given a relational structure and the content attributes (Taskar et al., 2007). Roughly speaking an RMN specifies the cliques and potential functions between attributes of related entities at template level. A single RMN model provides a coherent distribution for any collection of instances from a given relational schema.

Formally, a Relational Markov Network is defined as follows:
A Relational Markov Network $M = (C, \Phi)$ specifies a set of clique templates C and corresponding potential $\Phi = \{\phi_C\}$ to define a conditional distribution:

$$P(I.x|I.y, I.r) = \frac{1}{Z(I.x, I.r)} \prod_{C \in \mathbf{C}} \prod_{c \in C(I)} \phi_C(I.x_c, I.y_c) \tag{2.3}$$

where $Z(I.x, I.r)$ is a normalising partition function, $I.x$ is the set of the parent variable from a given entity and $I.r$ is the set of parent variables from entities related to a given entity.

$Z(I.x, I.r)$ is given by

$$Z(I.x, I.r) = \sum_{I.y} \prod_{C \in \mathbf{C}} \prod_{c \in C(I)} \phi_C(I.x_c, I.y_c) \tag{2.4}$$

To the best of the researcher's knowledge, RMN is the only rule-based undirected model found in literature. There was no any work found on structure learning in RMN. The learning discussed in (Taskar et al., 2007) only covered parameter learning assuming the clique templates are given.

One of the major weaknesses of RMN is that it does not provide a language for defining features. As a result RMNs require a feature for every possible state of a clique, making them exponential in clique size and limiting the complexity of dependencies they can model. Since there is no any defined method for revising the structure of RMNs, this framework was not considered for testing the solution proposed in this study for rational Belief Change in FOPL.

2.8 Directed First Order Probabilistic Logic

Directed models use Bayesian Networks as the basis for knowledge representation. Bayesian Networks provide a means of parsimoniously expressing joint probability distributions over many interrelated hypotheses.

2.8.1 Bayesian Networks

A Bayesian Network consists of 2 major components; the Structure and set of local probability distributions. The structure of a Bayesian network is a set of nodes representing random variables and directed edges between them for a Directed Acyclic Graph (DAG). The DAG represents direct qualitative dependence relationships between variables. A random variable denotes an attribute, a feature or a predicate whose value we may not be certain of. Each random variable has a set of mutually exclusive and collectively exhaustive possible values. That is, exactly one of the possible values is or will be the actual value, and we are uncertain about which one it is. The local distributions represent quantitative information about the strength of those dependencies. Together the graph and the local distributions represent a joint distribution over the random variables denoted by the nodes of the DAG.

Formally, a Bayesian Network is defined by:

1. a directed acyclic graph (DAG), $G = (V, E)$, where V is the set of vertices representing n discrete random variables $X = \{X_1, X_2, X_3, ..., X_n\}$, and E is the set of directed edges corresponding to conditional dependence relationships among these variables.

2. A set of local probability distributions: $\Theta = \{\Theta_1, \Theta_2, \Theta_3, ..., \Theta_n\}$, where each $\Theta_i = P(X_i|Pa(X_i))$ denotes the conditional probability distribution (CPD) of each node X_i given its parents in G denoted by $Pa(X_i)$.

Assuming conditional independence, the joint probability distribution for a Bayesian Network factorises in a product of the local distribution as shown in Equation (2.5)

$$P(X_1, X_2, X_3, ..., X_n) = \prod_{i=1}^{n} P(X_i|Pa(X_i)) \tag{2.5}$$

Equation (2.5) is called the chain rule for Bayesian Networks. It provides a method for determining the probability of any complete assignment to the set of random variables.

First Order Probabilistic logic based on Bayesian Networks does not allow cyclic relationships between variables, which often causes problem when dealing with domains that have bidirectional relationships, which are not necessarily causal.

2.8.2 Rule-based Directed Models

Rule-based Directed Models extend FOL logic with Bayesian Networks semantics. Examples of such languages include; Stochastic Logic Programs (SLP) (Muggleton, 1996),Bayesian Logic Programs (Kersting & Raedt, 2001), Relational Bayesian Networks(RBN) (Jaeger, 1997), Logical Bayesian Networks(LBN) (Fierens et al., 2005),etc. These solutions generally assign some probabilistic weights to conditional FOL statements.

Consider, for a example, the relationship between papers, papers they cite, and their categories, adopted from the CORA dataset. Figure (2.1) shows the propositional clauses encoding the relationship between three (3) papers, P, $P1$, $P2$.

Now, consider a Bayesian representation of the relationship between the papers

1. $class\_label(P2)$
2. $cites(P1, P2))$
3. $cites(P, P1)$
4. $class\_label(P1) \leftarrow class\_label(P2), cites(P1, P2)$
5. $class\_label(P) \leftarrow class\_label(P1), cites(P, P1)$

Figure 2.1.: Propositional Logic clauses for a domain with 3 papers, P, $P1$, and $P2$

shown in Figure (2.2). The Figure shows a repeated graphical sub-structure between the random variables; $class\_label(.)$, and $cites(.)$ highlighted by the unshaded ovals in Figure (2.2). The graphical structures and their associated conditional dependencies for the two(2) sub-structures are controlled by the same intensional regularities, but these regularities are not and can not be captured at propositional level. The approach rule-based models take is to upgrade these propositional clauses encoding the structure of a Bayesian Network to First Order clauses. Such a representation will, after introducing variables X and Y to stand in as place holders for the papers, result in the following FOL clauses; $class\_label(Y)$, $cites(X, Y))$, and $class\_label(X) \leftarrow class\_label(Y), cites(X, Y)$ shown in Figure (2.3).

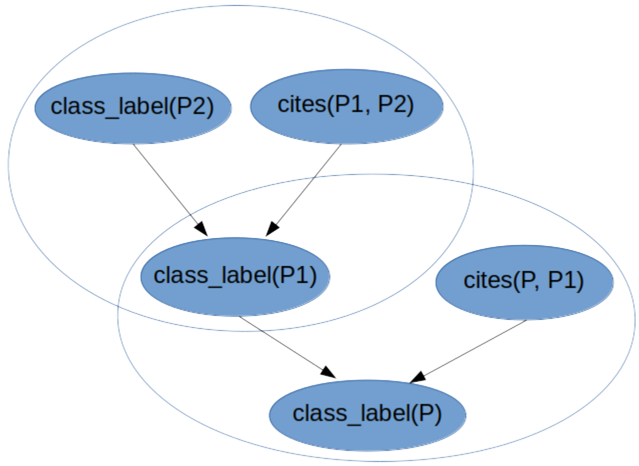


Figure 2.2.: Graphical View of the Propositional Logic clauses.

1. $class\_label(Y)$
2. $cites(X, Y))$
3. $class\_label(X) \leftarrow class\_label(Y), cites(X, Y)$

Figure 2.3.: First Order Logic clauses

In Bayesian Logic Programs (BLPs), the logical clause $A \leftarrow B_1, B_2, ..., B_n$ is expressed by what the authors called a Bayesian clause as follows:

A Bayesian Clause of the form $A|B_1, B_2, ..., B_n$ is an expression defined from a logical clause $A \leftarrow B_1, B_2, ..., B_n$. $n \geq 0$ and when $n = 0$ a Bayesian clause is called a Bayesian fact and is simply expressed as A. Figure (2.4) shows an example Conditional Probability table for a BLP.

2.8.3 Frame-based Directed Models

Most FOPL frameworks are in this category. Examples include; Probabilistic Relational Models (PRM) (Getoor, Friedman, et al., 2001), Probabilistic Entity Relation Models (Heckerman, Meek, & Koller, 2004), Multi Entity Bayesian Networks

$class\_label(Y)$	$cites(X,Y)$	$P(class\_label(X) \mid cites(X,Y))$
Reinforcement learning	1	(0.9, 0.02, 0.03, 0.025, 0.025)
Theory	1	(0.02, 0.8, 0.03, 0.025, 0.025)
...	...	...
Theory	0	(0.2, 0.2, 0.2, 0.2, 0.2)
...	...	...

Figure 2.4.: BLP Conditinal Probability Table

(MEBN)(Laskey, 2008), and Bayesian Logic (BLOG) (Milch et al., 2005). These models extend Bayesian Networks with concepts of objects, their properties and relationships between them. In a way, Frame-based Directed models are to Bayesian Networks what FOL is to propositional logic. Generally, the models define coherent formal semantics in terms of a probability distribution over sets of FOL interpretations. Given a set of ground objects, the model specifies a probability distribution over sets of interpretations involving these objects.

Frame-based Directed models have three(3) major components, and these are: the logical description of the domain, a probabilistic graphical model template, and a set of conditional probability distributions that corresponds to the graphical probabilistic model template. The logical description captures the FOL aspects of the representation. The probabilistic graphical model template describes the probabilistic dependencies in the domain, and the conditional probability distribution shows the statistical regularities for different instantiations of the model.

Among the afore-mentioned Frame-based Directed Models, MEBN has the advantage of being able to express arbitrary quantified FOL sentences and support recursion (da Costa, 2005). It also, unlike other frameworks, does not use object types as the unit of expression. In MEBN, distributions are specified over conceptually meaningful clusters of related hypothesis. This unit of representation facilitates flexible modular specification of the knowledge representation, that is not confined to the object types (Laskey, 2008).

This study uses the MEBN framework as a platform for the Proof of concept, hence this framework was used in this section to illustrate Frame-based Directed Models. A more detailed discussion of MEBN will be given in Chapter 7. Figure 2.5 shows an example graphical representation of MEBN Theory for the CORA dataset [1]. Figure 2.6 shows the FOL statement extracted from the MEBN Theory in Figure 2.5. Figure 2.7 shows an excerpt from the CPT of the class_label variable of the MEBN theory in Figure 2.5

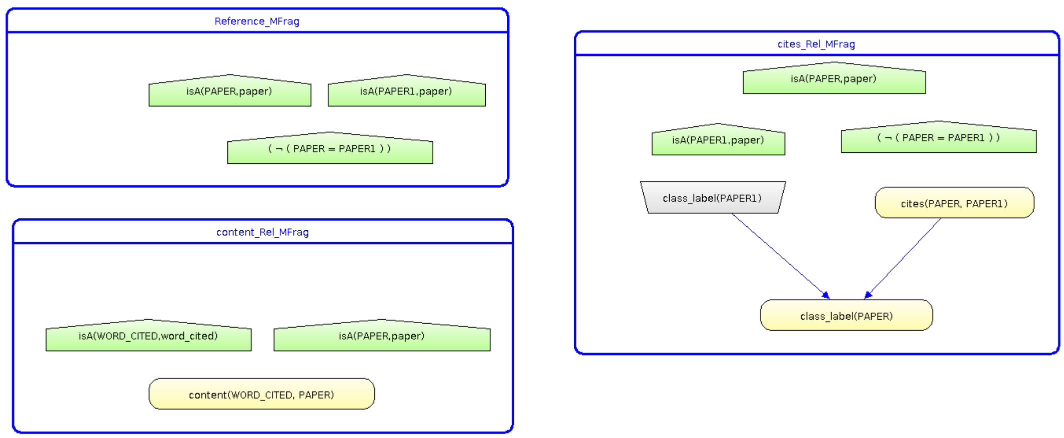


Figure 2.5.: MEBN Theory generated from the CORA dataset.

---

1. $isA(PAPER, paper)$
2. $isA(PAPER1, paper)$
3. $\neg(PAPER = PAPER1)$
4. $isA(WORD\_CITED, word\_cited)$
5. $content(WORD\_CITED, PAPER)$
6. $\forall PAPER,\ \exists PAPER1 :\ (class\_label(PAPER1) \wedge cites(PAPER, PAPER1) \vDash class\_label(PAPER))$

Figure 2.6.: FOL statements extracted from the MEBN Theory in Figure 2.5

[1]https://relational.fit.cvut.cz/dataset/CORA

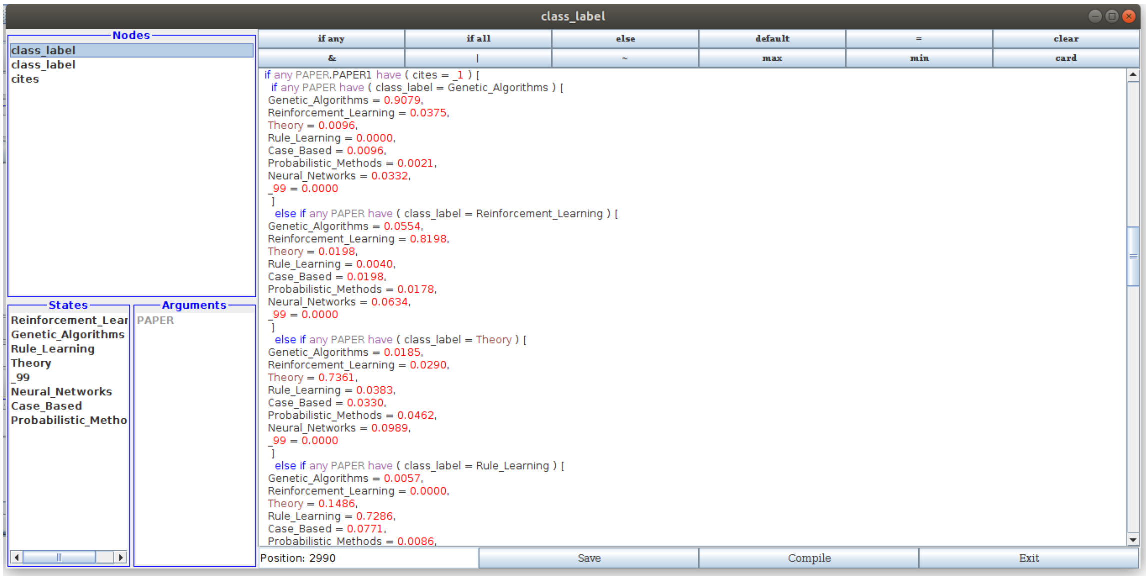


Figure 2.7.: An Except from the CPT for the variable class_label(PAPER).

Owing to the fact that MEBN is more expressive, in terms of being able to represent quantified FOL and handling recursion, compared to the other Frame-based Directed Models, this study uses MEBN as the Proof of Concept platform. It is, however, important to note that Rule-based Undirected Models, such as Markov Logic Networks, are more expressive than Frame-based directed model. In fact, Markov Logic Networks are believed to generalise all First Order Probabilistic Logic models. Ideally it would have been prudent to evaluate the Belief Change Model proposed in this study with such expressive representation, but owing to the complexity of the structure learning problem for such models, this research opted to evaluate the proposed solution with frameworks with well studied structure learning algorithms.

2.9 Conclusions

This chapter has established the need for a Knowledge Representation that inherently handles support for uncertainty. The discussion, started off with a characterisation of Open and Dynamic Computing Environments. Based on the literature surveyed, a desiderata for knowledge representation in Open and Dynamic Computing environments was established: (i) The knowledge representation should be expressive enough to represent knowledge about entities that are related to each other and to reason about the knowledge in the presence of uncertainty and increasing complexity; (ii) The KR should be capable of being easily integrated with a rational and objective operator for Iterated Belief Change; and (iii) The KR framework should support situational reasoning. Our investigation showed that frameworks that have emanated from the field of Statistical Relational Learning meet this desiderata. The only limitation they have is that there has not been any effort in making sure that the learning algorithms adhere to the principles of rational Belief Change.

Rational Belief Change is a well studied subject in classical logic. However classical logic has well defined boundaries between the ontological part of knowledge representations and the epistemological aspect of the knowledge. In Probabilistic Knowledge representation, probability is taken as part of the knowledge representation. However, in philosophy, probability is not ontological. It is thought of as an epistemological convenience that enables knowledge acquisition in the presence of uncertainty. This philosophical clumsiness in FOPL enables them to inherently handle Belief Change. This study will, in Chapter 4, take advantage of this philosophical clumsiness to define a Belief Change Model for First Order Probabilistic Logic.

A literature survey of existing work on Statistical Relational Learning showed that there are two major efforts that are converging into the field of First Order Probabilistic Logic for knowledge representation in environments dogged by uncertainty. One effort seeks to extend First Order Logic with probability and the second seeks to extend probabilistic graphical models with First Order Logic. Markov Networks and Bayesian Networks are often used to encode the conditional independence assumption to make inferences in such models intractable. FOPL frameworks that are based on Markov Networks are said to be undirected models and those based on Bayesian Networks are said to be directed. Although there are fewer frameworks based on undirected models, undirected models are generally believed to be more expressive than their directed counterparts. Markov Logic Networks are believed to be rich enough to be applied in all cases where directed models can be used (Richardson & Domingos, 2006). However, owing to the the complexity of structure learning in undirected graphical models and lack of tried and tested structure learning algorithms, this thesis developed a Belief Change operator for Directed Frame-based models. The proposed Belief Change meta-model is however generic enough to cater for Belief Change in the structure of any FOPL framework. Although the meta-model is a Bayesian model, it is not prescriptive on the nature of the underlying frame-based probabilistic logic it can be applied to.

Belief Change for Rule-based FOPLs is outside the scope of this work. Rule-based FOPL frameworks are usually weighted FOL statements, and probabilistic graphical models are only considered when the logic has been instantiated in order to enable inferences. Learning of the FOL statements is usually done through Inductive Logic Programming (ILP) (Muggleton, 1991). Nothing much have been done in literature on how ILP relates to rational Belief Change. One effort that investigates this is the work by Pagnucco and Rajatratnam (2005) which investigated Inverse resolution as Belief Change operator.

In view of the foregoing, this thesis concentrated on Belief Change on Frame based FOPL frameworks. The Multi-Entity Bayesian Networks (MEBN) framework was chosen for our Proof of concept. The developed operator can be used for any Directed Frame-based FOPL. The next chapter, Chapter 3, will discuss Belief Change in classical logic with the aim of establishing some knowledge nuggets that will be used in developing a Belief Change model directed First Order Probabilistic logic.

3. BELIEF CHANGE IN DYNAMIC DOMAINS

"Anything that gives us new knowledge gives us
an opportunity to be more rational."
Herbert Simon.

3.1 Introduction

Underlying the Belief Change problem is the quest for rationality in the process of changing beliefs about a given world in response to new information. This chapter is going to discuss the view of rationality that all arguments, with respect to Belief Change, to be presented in this thesis are based on. Although the discussion starts off with the view of rationality from a classical qualitative Belief Change perspective, the discussion will later digress to discussing rational Belief Change in First Order Probabilistic Knowledge representation. The perspective on Belief Change in FOPL presented in this chapter builds on the techniques that have come out of the research in classical Belief Change. Belief Change as a field of Artificial intelligence seeks to understand how software agents should rationally change their beliefs when they perceive new information or some information inconsistent with their beliefs. This assumes that the agents will have some prior set of beliefs that they need to change in response to some observations. The goal of the agents is therefore synonymous with the goal that human beings seek to achieve when they do science. From a philosophy of science standpoint, a product of science is a theory about the laws of nature that matches the results of experiments and observations. So, these three elements: theories, laws of nature, and experiments or observations, are not only complementary to science, but they are also the foundations upon which science is built. Science presumes that there are some premises that do not need to be verified or checked to start with. That is, there should be some beliefs about the world that should be accepted before any scientific enquiry. One of these fundamental beliefs is the belief in the existence of objective reality that is perceived more or less the same by everyone. This assumption is the fulcrum of science and knowledge, and science often assumes the objectivity of observed data.

From observing how human beings do science, it can be concluded that science has a logical framework, albeit that the framework reaches beyond deductive logic into the murkier realms of inductive reasoning and statistical inference. These realms are dogged by uncertainty and ambiguities, which human beings have over the years developed some techniques for handling. In AI, Belief Change has only been considered from a classical logic perspective. This makes it impossible for machines to use Belief Change in its current state to do science. This thesis argues that what machines lack to be able do science are structures to support a Logical Framework for science. As presented in Chapter 2, FOPL knowledge representation provides knowledge representation that enables such a logical framework. Belief Change in classical logic has a potential of providing rational operators for refining theories based on observation, but classical logic is too brittle to enable automatic evolution of knowledge bases. This Chapter presents an argument that techniques that have emanated from classical Belief Change can be extended to create a Logical Framework that can be used by machines to do science.

3.2 Belief Change: Preliminaries

The debate on Belief Change has traditionally been dominated by two types of Belief Change which have been coined Belief Revision and Belief Update (Friedman & Halpern, 1994). Both Belief Revision and Update attempt to capture the intuition that, given an observation of a new belief, minimal changes should be made to the 'Belief Set' (a set of beliefs currently held by an agent) in order to accommodate the new belief. The difference between them is that Belief Revision attempts to decide what beliefs should be discarded to accommodate a new belief, while Belief Update attempts to decide which changes occurred in the world that led to the new observation. Before discussing these types of Belief Change the discussion will start with a discussion of the basic principles underlying Belief Change.

3.2.1 Representation of beliefs

Given a sentence in a given language, $\mathcal{L}$, a rational agent should display three doxastical attitudes towards the sentence (Chhogyal, 2015):

1. a *belief*: is a sentence that the agent accepts as true
2. a *disbelief*: a sentence that the agent accepts as false
3. a *non-belief*: a sentence that the agent nether believes or disbelieves. That is the agent is agnostic of the sentence.

There are two major views to representation of beliefs:(i) the Belief Sets view and (ii) the Possible Worlds view.

In the Belief Sets view, the sentences of a formal language are taken as the objects of belief. An agent's beliefs are represented by a Belief Set K, which consists of sentences from some language $\mathcal{L}$. Negation of any of the sentences in the Belief Set gives a disbelief. All sentences that are not in the Belief Set, which are not negations of the sentences in the Belief Set are non-beliefs.

In the Possible Worlds view, propositions(sets of possible worlds) are the objects of belief. A Possible World is a candidate interpretation of a given logical sentence. The Possible Worlds view views the world to be in one of the many possible states at any given time. A state is one particular configuration of the different 'variables' in the domain. Each state is called a possible world. The Belief State of an agent is a set of all possible worlds that the agent believes one of them is the real world. To distinguish between the Belief Set K from the Belief State, $[K]$ is used to denote the Belief State. If a sentence α is true in all worlds in $[K]$, then it is said the sentence α

is a belief in $[K]$. For any sentence α, the set of all possible worlds in which α is true is denoted by $[\alpha]$. The terms *α-world* and *K-world* are used to respectively refer to a world where α is true, and a world where all the sentences in the Belief Set K are true.

In classical logic, where Belief Sets are used to model beliefs, an agent is uncertain about the truthfulness of a sentence when it is a non-belief relative to the agent. The bigger the set of non-beliefs is, the more uncertain the agent is about the truthfulness of any sentence. Probabilistic representations of beliefs use the Possible Worlds view which is based on the concept of Belief States. When Belief States are used, an agent is uncertain about the possible worlds in the set of worlds that satisfy the Belief Set. The bigger the set of possible worlds, the more uncertain the agent is. If the set of K-worlds has only one world, then the agent has no uncertainty about the world.

3.2.2 Probability as a Degree of Belief

In both the Belief Sets and the Possible Worlds representations, the objects of belief that the agent is uncertain about are lumped together into a single group and there is no way of comparing the relative degrees of belief on the objects within a group.

If probability is to be used as measure of degree of belief in the Belief Sets representation, an epistemic agent assigns a degree of belief of 1 to a sentence it is certain that it is true. That is, the degree of belief assigned by an agent to what it is certain of is 1. Owing to the dual nature of beliefs, the agent will assign a degree of belief of 0 to all disbeliefs. The relative degrees of belief on the non-beliefs will be greater than 0 but less than 1.

In the Belief Sets view to representation of beliefs, probability can be used as a mechanism for comparing the relative degrees of belief of logical sentences as follows: Assume that given a sentence α from a logical language $\mathcal{L}$, an agent assigns a numerical value to it to give relative degree of belief in its truthfulness. Let this assignment be represented by a function, P, that take a sentence from $\mathcal{L}$. Using the Kolmogorov (1956) laws of probability, P is a probability function if,

1. $0 \leq P(\alpha) \leq 1$
2. $P(\top) = 1$
3. for two sentences α and β, if α and β are logically disjoint, $P(\alpha \vee \beta) = P(\alpha) + P(\beta)$

where $\vee$ denotes a logical disjunction and $\top$ denotes a logical truth. P is referred to as the probability distribution over $\mathcal{L}$. $P(\alpha)$ is called the probability of α. The probability funtion P is therefore a means by which an agent can assign degrees of beliefs to sentences in $\mathcal{L}$. The first condition states that the probability of any sentence should be between 0 and 1. The second condition says that if the sentence is a logical truth it is assigned a probability of 1. The third condition states that if two sentences cannot be true at the same time, the probability of their 'combination' is the sum of their probabilities (i.e. $P(\alpha \vee \beta) = P(\alpha) + P(\beta)$).

A Belief Set K is said to be associated with a probability function if and only if every belief α in K is assigned a probability of 1, $P(\alpha) = 1, \alpha \subseteq K$. The set of all sentences such that $P(\alpha) = 1$ is known as the top of a probability function and it produces a Belief Set. However, P is not the only probability function that produces the Belief Set K. This is called the Non-uniqueness Problem (Lindström & Rabinowicz, 1989). It is possible that there may be another probability function that produces the same Belief Set. This, however, is not a problem for the conceptualisation presented in this thesis, since the reverse is not true. For any given P-function there is a unique Belief Set.

In classical logic, propositions are viewed as having a certain set-theoretic structure. This view enables representation of propositions as a set of Possible Worlds in which the propositions are true. As discussed in Chapter 2, the possible worlds view provides a logical semantics for integrating probability and logic.

In the possible worlds representation, degrees of belief are modelled on Possible Worlds. The language of possible worlds is more intuitive and more natural to probability theory. The agent's uncertainty is about which among all the possible worlds is the actual world in the current time slice. The set of propositions, defined as a set of sets of Possible Worlds, is an algebra $\mathcal{A}$ over a non-empty set of possible worlds $\mathcal{W}$. An algebra $\mathcal{A}$ is a set that contains $\mathcal{W}$ and is closed under complementation and finite intersections. Thus, if a proposition A is in $\mathcal{A}$, then $\neg A$ is also an element of $\mathcal{A}$. Closed under finite intersection implies that if $A_1, A_2, A_3..., A_n$ are elements of $\mathcal{A}$, then $A_1 \cap A_2 \cap A_3 \cap ... \cap A_n$ is an element of $\mathcal{A}$.

More formally, let an Algebra, $\mathcal{A}$, over a set of all possible worlds $\mathcal{W}$ be defined as follows:

Definition 3.1 : *$\mathcal{A}$ is an algebra over $\mathcal{W}$ iff $\mathcal{A} \subseteq \mathcal{P}(\mathcal{W})$, the power set of $\mathcal{W}$, such that given proposition A and B (Spohn, 2012)*,

a. $\mathcal{W} \in \mathcal{A}$

b. if $A \in \mathcal{A}$, then $\overline{A} \in \mathcal{A}$

c. if $A, B \in \mathcal{A}$, then $A \cup B \in \mathcal{A}$

d. for each countable $\mathcal{B} \subseteq \mathcal{A}$, $\bigcup \mathcal{B} \in \mathcal{A}$. That is $\mathcal{A}$ is a sigma algebra.

The best developed account of degrees of belief is the theory of subjective probabilities (Huber, 2009). In this view, degrees of belief follow the laws of probability and are governed by Kolmogorov (1956) laws of probability. A function $P : \mathcal{A} \rightarrow \mathcal{R}$ from the algebra $\mathcal{A}$ to a set of real numbers $\mathcal{R}$ is a probability on $\mathcal{A}$ if and only if:

1. $P(A) \geq 0$

2. $P(\mathcal{W}) = 1$

3. $P(A \cup B) = P(A) + P(B)$, if $A \cap B = \emptyset$

The triple $(\mathcal{W}, \mathcal{A}, P)$ are referred to as the probability space. Such a probability space defined over possible worlds provided the basis for the definition of an epistemic space over possible Bayesian Networks defined in Chapter 4.

3.2.3 Belief Change Principles

Research in Belief Change, customarily starts off with a characterisation of the commitments a rational Belief Change function should satisfy. These commitment are usually presented as a set of postulates that a Belief Change function should satisfy. Flouris, Plexousakis, and Antoniou (2006) identified a partial list of six principles that these postulates are drawn from. Following are the principles:(i) primacy of new information; (ii) irrelevance of syntax; (iii) consistence maintenance; (iv) Fairness; (v) adequacy of representation; and (vi) minimal change

Principle of Primacy of New Information: It is usually assumed that observations are observed with certainty and common intuition dictates that newer information generally reflects a newer and more accurate view of a domain. It is against this argument that the principle of primacy of new information is derived. A representation of knowledge about a given domain must therefore be able to accept the new information unconditionally. However, owing to the pervasiveness of uncertainty in open and dynamic computing environments (especially the web) this principle may need to be relaxed. In classical Belief Change, a lot of work has been done under the auspices of non-prioritised Belief Change (Hansson, 1999; Meyer, Ghose, & Chopra, 2001). Non-prioritised Belief Change allows new information to be rejected partially or totally. This is one area where probabilistic knowledge representations have an advantage over deterministic ones since they have principled mechanisms for handling uncertainty.

Principle of Irrelevance of Syntax: First Order logic is known to be very flexible in the specification of axioms. This has raised worries about whether the Belief Change operation will not be affected by the syntactical representation of the Knowledge Base or the new information. This gave birth to the principle of irrelevance of syntax. It states that "revision of two logically equivalent knowledge bases formed using completely different axioms should result in two logically equivalent knowledge bases regardless of the differences in syntax". This principle is more prone to violation in Belief Bases under the foundational view point. In the foundational approach different sets of axioms imply different justifications for some axioms.

Principle of Consistence Maintenance: Anything can be inferred from an inconsistent Knowledge Base. As such, under whatever circumstances, revision of a Knowledge Base should result in a consistent Knowledge Base, otherwise there is no point in revising the Knowledge Base. The principle of consistency dictates that the Belief Change operator should not result in an inconsistent Knowledge Base. However, the term consistence has been used to refer to different things in knowledge representation and database technology. The augments for consistency still stand though,regardless of how consistency is defined. Consistency should be maintained after revision/update of a Knowledge Base.

Principle of Fairness: The principle of fairness guarantees objectivity and reproducability of the result of a Belief Change. A Belief Change operator should be objective enough to reproduce the same result if a revision by the new information is repeated on the same knowledge base at any time. This is quite a challenge in cases where the Belief Change operation is intractable and heuristics have to be used. In such cases the operator might converge to one local optimal solution at one time and to another local optimal solution at another time.

The Principle of Adequacy of Representation: The knowledge representation language should be rich enough to represent the KB , the new information and the resultant KB.

Principle of Minimal Change: Rationality is widely believed to be economical with what is already believed. That is the revised KB should conserve as much of what was believed before revision as possible. This is known as the principle of minimal change. However, the challenge is on how *minimal change* is to be defined.

In classical Belief Change there are two main approaches to modelling rational Belief Change; the Foundational Theory and the Coherence Theory. In the Foundational Theory one needs to keep track of the justification for one's beliefs. Propositions that do not have justification should not be accepted as beliefs. That is, according to this theory, revision of a Belief Set by a sentence, α, should consist of first giving up all beliefs that will no longer be satisfactorily justified if α is accepted in to the Belief Set, then adding all the beliefs that are justified by the introduction of α. The Truth Maintenance System (TMS) (Doyle, 1979) is a the flagship realisation of the foundational approach to Belief Change. The TMS is a system for keeping track of justifications in Belief Revisions. As such, whenever some beliefs have to be given up the TMS is consulted to check which beliefs are not justified.

In the coherence theory, the aim is to maintain consistency in the Belief Set revised by α, whist ensuring minimal changes. In the coherence theory, beliefs are accepted on the basis of how coherent they are with what is already believed. Although the coherence theory does not directly provide justification for any beliefs held, justifications for a belief are provided holistically. Whether a belief is believed depends on how well it fits together with everything else that is believed. As can be seen so far the key principles of the coherence theory are consistency maintenance and minimal change.

The foundational theory has severe problems handling logical relations between beliefs. As a result the coherence theory is usually preferred. This, coupled with the fact this thesis is investigating Belief Change in Frame-based FOPL dictates that we consider the coherence view to Belief Change. The focus of the study will be primarily on the principle of minimum change and consistency maintenance. The discussion that follows will look deep into Belief Revision and Belief Update. In principle, both Belief Change types try to explain the source of incorrect beliefs at any given time but make different assumptions about the sources of incorrect beliefs.

3.3 Belief Revision

Belief Revision has been widely understood as the process of incorporating some new information about a static world in a Belief Set. In static worlds, belief changes are only necessitated by the fact that the system's beliefs about the world are mistaken or incomplete since there will be no changes in the domain. Belief Revision takes a coherence view of Belief Change, which advocates that credibility of an axiom depends on how coherent it is with other axioms in the Knowledge Base (KB). The premise behind the coherence view is, if a Belief Revision operation calls for some beliefs to be retracted (in order to keep the KB consistent after the Belief Change operation), the relative entrenchment of a belief depends on how coherent the belief is with other beliefs in the Belief Set. The most popular Belief Revision theory in qualitative Belief Change is the AGM theory (Alchourrón, Gardenfors, & Makinson, 1985). Given a Belief Set K, AGM theory defines a set of postulates that govern the process of revising K by new information α not present in K (for more on the AGM postulates see ((Alchourrón, Gardenfors, & Makinson, 1985),(Benferhat, 2010)).

Assuming a logically finite, classical propositional language, denoted by $\mathcal{L}$, consequence operation Cn, and revision operator $*$, the AGM postulates for Revision are as follows:

(R1) Closure: $K * \alpha = Cn(K * \alpha)$

(R2) Success: $\alpha \in K * \alpha$

(R3) Inclusion: $K * \alpha \subseteq K + \alpha$

(R4) Vacuity:If $\neg\alpha \notin K$, then $K * \alpha = K + \alpha$.

(R5) Consistency: $K * \alpha$ is consistent if α is consistent.

(R6) Extensionality: If $(\vDash \alpha \equiv \beta) \in Cn(\emptyset)$, then $K * \alpha = K * \beta$.

(R7) Superexpansion: $K * (\alpha \wedge \beta) \subseteq (K * \alpha) + \beta$

(R8) Subexpansion: If $\neg\beta \notin Cn(K * \alpha)$, then $(K * \alpha) + \beta \subseteq K * (\alpha \wedge \beta)$.

where α and β are some logical sentences such that $\alpha, \beta \in \mathcal{L}$.

Closure ensures that when a Belief Set is revised by α, the resulting Belief Set holds all the logical consequences of α. Success postulates ensures that revision by α should result in the acceptance of α. This speaks to the principle of primacy of new information. The inclusion postulate states that the resulting Belief Set of expanding K by α should at least hold all beliefs of the result of revising K by α. The consistency postulate ensures that if α is consistent, then revision of K by α should be consistent. Unfortunately, though the AGM postulates are sound in principle, they do not detect which beliefs are to be given up if the Belief Set is found to be contradictory to the new information. This has resulted in alternative Belief Revision mechanisms that are based on the concept of epistemic entrenchment. A belief is said to more entrenched in the Belief Set if agents are less willing to give it up. Less entrenched beliefs will be preferentially given up to accommodate new beliefs. The solutions that use this concept include the Entrenchment Relations (Grdenfors & Makinson, 1988), Ordinal Conditional Functions (OCF) (Spohn, 1988), Minimal Conditional revision (Boutilier, 1996). The basic principle in these solutions, is using some sort of ordering or ranking of the beliefs based on the agents' unwillingness to give them up. This ranking is used as a mechanism for deciding which beliefs should be given up to accept a new belief. Such a ranking of Beliefs is known as the *Epistemic State*, and it is a very important component for Belief Change operator. To shed more light on the semantics of epistemic states and how they are used for Belief Change, the discussion that follows focuses on Spohn's qualitative ranking (Spohn, 1988).

Assuming a fixed set of possible worlds, the OCF operator maps all worlds in into a set of natural numbers (ranks) based on the agents' willingness to give each one of them up. Such a ranking $\kappa : W \rightarrow N$ assigns to each world, a natural number reflecting its plausibility. If $\kappa(w) < \kappa(v)$, then w is more plausible or more consistent with the agents' beliefs than v. The maximally plausible worlds are given rank zero

(0) and the impossible worlds are given rank ∞. This ranking induces the Belief Set and a revision function. The Belief Set K is given by

$$K = \{\alpha \in \mathcal{L} : \kappa^{-1}(0) \subseteq ||\alpha||\} \tag{3.1}$$

where, $\mathcal{L}$ is logically finite propositional language defined over a set of sentences, α is an element of $\mathcal{L}$, and $||\alpha||$ is a set of $\alpha - worlds$, the elements of $\mathcal{W}$ satisfying α. Later in this chapter the symbol, α_{ji} is used to explicitly denote a sentence as an edge from node j to node i in a Bayesian Network. A Bayesian Network is interpreted to be a possible world. The edge α_{ji} is said to be entailed by the Bayesian Network B, denoted by $B \vDash \alpha$. The revision function takes the most plausible worlds that entail α as the epistemically possible worlds. Thus the revision function $min(\alpha, \kappa)$, produces the revised Belief Set K^*_α, defined as follows:

$$K^*_A = \{\beta \in \mathcal{L} : min(\alpha, \kappa) \subseteq ||\beta||\} \tag{3.2}$$

That is, a propositional atom is accepted in the revised Belief Set ($\beta \in K^*_\alpha$) iff $min(\alpha, \kappa) \subseteq ||\alpha||$. An analogical conceptualisation of this idea in quantitative Belief Revision is obvious. Probabilities can easily be used as a basis for ordering or ranking the plausible worlds. A discussion of the quantitative analogical conceptualisation Belief Revision will be presented in Chapter 4.

Apart from Belief Revision, there are two other Belief Change operations that are discussed along with Belief Revision. Original work on AGM identified two other belief operation; Belief Expansion and Belief Contraction.

Belief Expansion deals with a situation where a non-belief is made a belief. This is a situation where an agent acquires new information, which is not contrary to its current beliefs. Thus, given a belief α and a Belief Set K, the result of an expansion of K with α is denoted by K^+_α and in formal Belief Change terms is given by:

$$K^+_\alpha = Cn(K \cup \alpha) \tag{3.3}$$

Cn denotes the logical consequence of the resulting Belief Set.

Belief Contraction arises when the agent receives some new information inconsistent with the currently held beliefs (explicit beliefs and their logical consequences). Thus, Belief Contraction transforms beliefs into non-beliefs. This is more complicated than expansion. This is mainly because in Contraction, like Revision, there is no unique way of constructing the resulting Belief Set simply based on Information Economy.

Belief Revision is related to Belief Contraction and Belief Expansion through the Levi identity (Levi, 1977), which is defined as follows:

$$K_\alpha^* = (K_{\neg\alpha}^-)_\alpha^+ \tag{3.4}$$

3.4 Belief Update

A lot of work has been done on Belief Change in dynamic worlds (Belief Update), but the proposed approaches still remain less popular as compared to the approaches for Belief Revision. The work by Katsuno and Mendelzon (1991), known as the KM theory, on Belief Update has attracted considerable attention. The KM theory uses the event model to capture knowledge changes in a dynamic world. The KM theory takes observation of a new axiom as evidence that there have been a change in the world. If some new fact α is observed, update assumes that it is a result of some unspecified change in the world (i.e some event occurrence or action). The resulting KB is not known except for the fact that it accepts α. There are many possible candidate KBs, so the challenge is that of identifying the KB closest to the true representation of the knowledge about the domain. Katsuno and Mendelson (1991) proposed a general characterisation (postulates) of Belief Update that provides the constraints that must be satisfied to reflect the changes in the domain.

Using $\diamond$ as the update operator, Katsuno and Mendelzon (1991) defined the following postulates for Belief Update:

(U1) $KB \diamond \alpha \models \alpha$.

(U2) If $KB \models \alpha$, then $KB \diamond \alpha \equiv KB$.

(U3) If KB and α are both satisfiable, then $KB \diamond \alpha$ is satisfiable.

(U4) If $\models \alpha \equiv \beta$, and $KB_1 \equiv KB_2$, then $KB_1 \diamond \alpha \equiv KB_2 \diamond \beta$.

(U5) $(KB \diamond A) \wedge \beta \models KB \diamond (\alpha \wedge \beta)$.

(U6) If $KB \diamond \alpha \models \beta$, and $KB \diamond \beta \models \alpha$, then $KB \diamond \alpha \equiv KB \diamond \beta$.

(U7) If KB is complete, then $(KB \diamond \alpha) \wedge (KB \diamond \beta) \models KB \diamond (\alpha \vee \beta)$.

(U8) $(KB_1 \vee KB_2) \diamond \alpha \equiv (KB_1 \diamond \alpha) \vee (KB_2 \diamond B)$.

where α and β are some logical sentences such that $\alpha, \beta \in \mathcal{L}$.

If α, a result of some change in the world, is observed, one would like to consider all the possible explanations for what might have changed to make α true and choose the most plausible explanation for how the world may have changed to accept α. Katsuno and Mendelson (Katsuno & Mendelzon, 1991) proposed a set of preorders over the set of possible worlds, $\mathcal{W}$, $\preceq_w$: $w \in \mathcal{W}$. The relation $\preceq_w$ is a reflexive and transitive over $mathcalW$. The relation $u \preceq_w v$, means u is as at least as plausible a change as v relative to w. This intuitively means that, if at time t, w was believed to be the true state of the world, then the world is more likely to transition to world u than to v. The most plausible world after observation of α is therefore the minimal worlds with respect to w that result in acceptance of α.

Just as in the case of the AGM theory in revision, the KM theory does not dictate which beliefs are to be given up if the Belief Base is found to be inconsistent with the new information. To counteract this weakness the κ-rankings can also be adopted to define Epistemic States for the KM theory. Each world v is simply associated with ranking such that the set $\kappa^{-1}(0)$ is the set of the most plausible worlds and use $min(\alpha, \kappa_w)$ as the update function.

3.5 Unified Belief Change Models

If the world is dynamic, both Belief Revision and Belief Update can provide possible explanations for the misconceptions in the Knowledge Base. This has prompted research efforts aimed at unifying Belief Revision and Belief Update in one Belief Change operator. Examples of such works include (Boutilier, 1998; Friedman & Halpern, 1998; Hunter & Delgrande, 2005; Shapiro & Pagnucco, 2004). The work presented in this thesis extends such work, and particularly the Belief Change model proposed by Boutilier (1998) by: (i) defining a Unified Belief Change Model based on probability as the measure of degree of belief; and (ii) enabling the model to be used for Belief Change in FOPL. Further discussion on this model is given in Chapter 4.

An investigation of literature revealed that there are two alternative ways through which Belief Update semantics can be combined with Belief Revision semantics to enable reasoning about a dynamically changing system. Either,

1. Modelling epistemic states in a temporal space and then define state transition probabilities or,

2. modelling event plausibility semantics that can be used to explain state transitions.

Modelling epistemic states in a temporal space is the most widely used solution in modelling dynamical systems. It finds its origin in classical probability theory and it leverages upon the Markov assumption.

Definition 3.1: A Markov Chain (Kemeny & Snell, 1983) over states S_1, S_2, S_3, ... is a measure P on a set of worlds $\mathcal{W}$ such that:

i. $P(S_{n+1} = s_{n+1} | S_n = s_n, S_{n-1} = s_{n-1}, ..., S_0 = s_0) = P(S_{n+1} = s_{n+1} | S_n = s_n)$

ii. $P(S_{n+1} = a | S_n = b) = P(S_{m+1} = a | S_m = b), m \neq n$

The first requirement speaks to the "forgetfulness" of Markov processes. Thus, the probability of a transition from state $S_{n+1} = s_{n+1}$ to $S_n = s_n$ is independent of states preceding $S_n = s_n$ given $S_n = s_n$. The second requirement (the invariance assumption) states that the probability of a transition from one state to another is independent of the time of the transition. Although this approach is believed to be able to model most real world situations, this thesis argues that it is insufficient for modelling state transitions in evolving FOPL graphical structures. The evidence that's been observed so far has an effect on the transitions probabilities, hence they cannot be assumed to be constant overtime. Thus, the problem with this approach emanates from the second requirement in Definition 3.1. This calls for richer semantics to model the dynamics of the network structure as evidence is being observed. This is where solutions that use event probabilities become advantageous. The argument in event-based semantics for belief change is that events that occur in the domain provides an impetus for change in the domain. In solutions that include event semantics (e.g. (Boutilier, 1998; Jin & Thielscher, 2004; Lang, 2007; Shapiro & Pagnucco, 2004)) the transition probabilities are assumed to be dependent on occurrence of certain observable events in a given state. However, this cannot be directly adopted for evolution FOPL network structures since Network structure change events in Probabilistic Graphical Models are not observable. Fortunately, classical approaches to learning Bayesian Network structures have some techniques we can leverage upon to address this challenge. These techniques are discussed in Chapter 4.

Roughly, Belief Revision treats new information as evidence that the previous beliefs were incorrect, while Belief Update treats new information as evidence that the world has changed, hence what has been observed is the least that could have possibly changed. In reality, observation of new information may imply revision, update or both. This calls for a unified Belief Change model that caters for both Belief Revision and Belief Update. A lot of effort has been directed towards this goal for qualitative belief change (e.g (Boutilier, 1995; Friedman & Halpern, 1998; Lang, 2007), but not much has been done with respect to probabilistic Belief Change. The discussion that follows, focuses on how this is done in qualitative Belief Change and highlights techniques that can be adapted for the quantitative counterpart.

The major problem with Belief Revision is that in its original form it cannot handle change in dynamic environments rendering it unsuitable for real life applications and more specifically for this thesis' problem context. On the other hand, belief update in its original form does not allow inferring new beliefs about the past from latter observations. Unified Belief Change models are believed to address the afore-mentioned weaknesses of Belief Revision and Belief Update by making them complement each other.

Notable amongst the work on Unified Belief Change Models, is the work by Boutilier (1998). Boutilier (1998) assumed that events that occur in a given world provide an impetus for change in the domain and used plausibility distribution over events to model epistemic states for a belief update model. This formed the first step of the belief change model that captured update. To handle revision, Conditionalisation was used. Thus, a unified belief change model was modelled as a two-step process that involves update first and then followed by revision of the updated belief state conditioned on evidence. An indepth discussion of the Boutilier's Unified Belief Change Model will be given in chapter 4. The unified Belief Change model defined in this thesis builds on this model. The Belief Change Meta Model presented in chapter 4 is a lifted abstraction of Boutilier's model.

3.5.1 Iterated Belief Change

One of the key weaknesses of early Belief Change models is their inability to handle iterated belief change (Darwiche & Pearl, 1997). This makes it impossible for machines to be able to automatically evolve their knowledge bases as they acquire new information. The problem stems from the fact that these models, given a Belief State, an Epistemic State and some observation, are only able to give the Belief State and not the Epistemic State. As can be deduced from the foregoing discussion, it is the epistemic state that provides the guidance for changes in belief due to subsequent observations. Thus, to enable Iterated Belief Change the result of a Belief Change should give both the Belief State and the Epistemic State. What has been discussed so far deals with how the Belief State is changed and not the Epistemic State. As discussed in much of the related literature (e.g. (Boutilier, 1998; Darwiche & Pearl, 1997; Hunter & Delgrande, 2005)), the Belief Change operator should change the Epistemic State and the Belief State will be generated as set of maximal worlds from the Epistemic State. To understand the requirements for Iterated Belief Change, it is imperative that the relationship and the differences between the Belief State and the Epistemic State be highlighted. A Belief State characterises the set of propositions that the agent is committed to at a given time. An Epistemic State holds the relative entrenchments of all propositions relative to the current belief state conditioned on the hypothesised evidence. Effectively, a Belief Change operator will change these relative entrenchment when some evidence is observed and the Belief State will be the maximal set of worlds from the Epistemic State. In Quantitative Belief Change, the Epistemic State can be thought of as a probability distribution over all the possible worlds, and the corresponding relative propensities (given in terms of event probability distributions conditioned on possible worlds) of moving from one possible world to another.

3.6 Other perspectives to Belief Change in Dynamic Domains

There are other perspectives to Belief Change that have been discussed in literature. These include Imaging (Lewis, 1976), and Focusing (Dubois & Prade, 1997). Another concept which is also similar to solutions for Belief Change in dynamic domains is Concept Drift (Kadlec, Grbić, & Gabrys, 2011).

3.6.1 Imaging and Focusing

Imaging (Lewis, 1976) is the starting point for a lot of work that has looked at probabilistic revision. Imaging as introduced by Lewis (1976) is much closer to Belief Update than Belief Revision. The Imaging of a probability function P on the evidence $/alpha$ constitutes a new Belief State (probability function) obtained by shifting the original probability mass(relative entrenchment) of the worlds where $\neg\alpha$ is true($\neg\alpha - worlds$) over to the worlds where α is true ($\alpha - worlds$) that are assumed to be closest to the $\neg\alpha - worlds$. The probability mass is shifted to the worlds closer to the $\neg\alpha - worlds$ to ensure minimal change. There are however other works on belief change that rather shift the probability of the $\neg\alpha - worlds$ and the worlds close to them to the worlds that are as far from them as possible (e.g. (Rens & Meyer, 2015)). The intuition of such Belief Change operators is that the closer the world w is to the $\neg\alpha - worlds$ the less probable the world is. Thus, probability mass should therefore be shifted as well from w as we shift probability mass from $\neg\alpha - worlds$. However, such Belief Change solutions are more likely not to adhere to the principle of minimal change compared to Imaging as originally proposed by Lewis (1976).

Focusing (Dubois & Prade, 1997) seeks to make a distinction between two components of the body of knowledge: (i) generic knowledge and (ii) factual evidence. Dubois and Prade (1997) argue that belief revision deals with modifying generic knowledge when receiving new pieces of generic knowledge, while focusing deals with applying generic knowledge to the reference class of situations, which corresponds to all the available evidence gathered on the case under consideration. The difference between generic knowledge and factual evidence can be illustrated by a diagnosis problem. The generic knowledge of a clinician consists in his/her knowledge about the links between the diseases and the symptoms and the distribution of the diseases in the original population (in practice, the likelihoods and the prior probabilities). The factual evidence consists in the symptoms collected from the patient under consideration. Focusing then applies the knowledge about the the links between diseases and the symptoms to a specific situation focusing on the subset of beliefs that are within the current focus of attention. Thus, focusing is more of an inference solution rather than a Belief Change solution.

3.6.2 Concept Drift

The field of Concept Drift (Kadlec, Grbić, & Gabrys, 2011; Moreno-Torres et al., 2012; Žliobaitė, Pechenizkiy, & Gama, 2016) emanated from the machine learning community as a response to the observation that predictive performance of generative models degrade over time if the models assume static relationships between input and output variables. This is owing to the fact underlying relationships in the data change over time with the changes in the domain. The changes in underlying relationships in the data are a reflection of the changes in the environment the data is being emitted from. The goal of Concept Drift in machine learning is to deploy models that would self-diagnose and adapt to changing data over time (Žliobaitė, Pechenizkiy, & Gama, 2016). This goal is similar to what this thesis seeks to achieve. However, the solutions that have emanated from concept Drift are different to the goal of this research in the following respects: (i) The focus of Concept drift solutions is on prediction rather than an explicit representation of the belief state of the domain; (ii) The solutions give primacy to the new evidence and pays no attention to the principle of minimal. This means rationality in the evolution of the predictive models is not of interest in Concept Drift solutions.

3.7 Conclusions

This chapter discussed the researcher's findings from a literature survey of techniques for belief change in general with more emphasis on dynamic environments. This thesis takes a coherence view to Belief change. The coherence view presupposes that beliefs are removed or entrenched based on their coherence to other belief that are accepted. Belief Revision and Belief Update are the most widely studied Belief change solutions based on the coherence view. Belief Revision and Belief Update make different assumptions on the source of inconsistencies in Knowledge Bases. Belief Revision assumes that inconsistencies arise from incorrect propositions in a world that is assumed to be static, while Belief Update assumes that inconsistencies arise from changes in the domain. However, most real life applications of Belief Change involve both types of change. Thus, there has been a lot of work that seeks to create a unified Belief Change model that caters for both revision and update. The unified Belief Change Model developed in this study is based on one such model proposed by Boutilier (1998).

One of the key requirements to enable automatic evolution of knowledge bases in dynamic domains is support for iterated revision. This was one of the major limitations of the early Belief Change models. Research in Belief Change generally agree that for this to be possible there is a need that the result of Belief Change operation be an Epistemic State rather than a Belief State. As will be seen in Chapter 4, this thesis defined an epistemic state as tuple of two probability distributions; one over all plausible events given some hypothesised Belief State and the other over all plausible Belief States.

This chapter also discussed other views to Belief Change that are conceptually related to Belief Revision and Belief Update, but not necessarily based upon these concepts. These included Imaging, Focusing and Concept Drift. The Distinction between Generic Knowledge and Factual Evidence is very important to Belief Change if it is to be used as basis for a logical framework through which machines can do science. Belief Change necessitated by the temporal dynamics of a domain may only need to be used for inferences rather than change of the generic knowledge representation. This aspect is outside the scope of the work presented in this thesis.

4. THE UNIFIED BELIEF CHANGE MODEL FOR BAYESIAN NETWORK STRUCTURES

"Not to be absolutely certain is, I think, one of the essential things in rationality."
Bertrand Russell

4.1 Introduction

This chapter presents the Unified Belief Change Model for Bayesian Network based knowledge Representations in open and dynamic computing environments. If Bayesian Network based FOPL is used for knowledge representation in open and dynamic domains as proposed in Chapter 2, there is a possibility of observations becoming inconsistent with the underlying Bayesian Network structure owing to the dynamic nature of the domain. In such cases the network structure needs to be rationally evolved to reflect the changes in the domain (Belief Update), and to correct incorrect beliefs about the domain (Belief Revision).

Even though some work has been done on structure learning in Bayesian Network based First order knowledge representation (e.g. (Getoor, 2000), (Natarajan, Wong, & Tadepalli, 2006), (Kersting & Raedt, 2008), (Coutant, Leray, & Le Capitaine, 2014) (Ettouzi, Leray, & Messaoud, 2016)), the focus has been on how the network structure can be automatically learnt from data, given some prior knowledge about the structure. In these solutions, the emphasis is on estimating the network structure that will then be assumed to be constant thereafter. If the structure becomes inconsistent with the observations, the structure has to be re-learnt from the observations emitted from the domain. As a result, such solutions do not consider the problem from a

Belief Change perspective where the structure can be automatically evolved as the data inconsistent with the current network structure is observed.

In classical logic, the problem of rationally capturing and effecting changes in beliefs about the domain is addressed by techniques that have emanated from the field of Belief Change. Belief Change techniques seek to provide mechanisms for evolving knowledge representation in a rational manner. The solution to Belief Change formulated in this chapter draws inspiration from the work that has been done on rational Belief Change in classical logic.

4.2 Overview of the Proposed Belief Change Solution

As highlighted in most research works on learning Bayesian-based First Order Probabilistic models (e.g. (Getoor, 2000; Natarajan, Wong, & Tadepalli, 2006)), structure learning in First Order Bayesian models is fundamentally the same as that of propositional Bayesian Networks (Getoor, 2000). Hence, a solution to the research problem being investigated in this study for propositional Bayesian Networks can be easily extended to First Order Probabilistic logic. Hence, the discussions of the problem and the solution hence forth are going to be in the general context of Belief Change in propositional Bayesian Networks. At this point, it is also important to highlight that the Unified Belief Change Model formulated in this work builds upon the significant progress that has been made on Bayesian Network structure learning in propositional Bayesian networks. Researchers have over the years made significant progress in formalising the theory of Bayesian structure learning (Benferhat, 2010; Cooper & Herskovits, 1992; Friedman & Goldszmidt, 1997; Neapolitan, 1990) and development of efficient algorithms for learning Bayesian structure using prior knowledge (Chickering, Heckerman, & Meek, 1997; Eggeling et al., 2019; Heckerman, Geiger, & Chickering, 1995; Liu et al., 2018; Zhao et al., 2015). Notwithstanding these substantial advances in Bayesian structure learning, there has been no focus on whether these structure learning algorithms adhere to the principles of rational

Belief Change. Usually, as noted in (Benferhat, 2010; Dubois, 2008), Belief Change in Bayesian networks is usually thought of as simple propagation of probability mass in the network as a result of observations consistent with the dependencies encoded in the Bayesian Network structure.

However, in dynamic domains observations are not always bound to be consistent with the Bayesian network structure. This calls for a mechanism for Belief Change on the Bayesian network structure when evidence inconsistent with the structure is observed. This is very important if Bayesian networks are to be the underlying formalism for knowledge representation. This work therefore contributes toward addressing this issue by deriving a Belief Change Model for Bayesian network structures representing knowledge in dynamic domains. This thesis argues that for both Belief Update and Belief Revision to be captured, there is a need for a mechanism for handling event semantics as is done in classical Belief Change. The events are believed to be the ones that provide an impetus for change in the domain (Boutilier, 1998). From this assumption, a knowledge representation of a given domain is modelled as a dynamical system and a dynamic Bayesian network is defined to capture the dynamics of the knowledge representation. This highlights the first contribution of this thesis, a Belief Change meta-model, which forms the basis of the Unified Belief Change Model defined in this chapter.

In most solutions for modelling dynamic systems, the world is assumed to change owing to some observable actions/events. In modelling evolution on Bayesian network structure, the observations and the events that may have caused a change in the domain are unobservable. This implies that solutions, such as Partially Observable Markov Decision Processes (POMDP), which assume that changes in the domain are a result of observable ontic actions/events cannot be directly adopted for modelling Belief Change in Bayesian Network structure.

The Belief Change Model derived in this chapter is validated by analogy with two classical Belief Change theories. To the best of the author's knowledge, the work presented in this thesis is the first attempt to consider Bayesian structure learning from this perspective. The goal in this endeavour is to argue the possibility of using Belief Change principles to solve the structure learning problem in First Order probabilistic knowledge representation and the efficacy of such an approach.

4.3 Preliminaries

This section introduces the problem that this work is addressing using a motivating example. The section also discusses the basic concepts this work builds upon, some notation, and definitions from the fields of Belief Change and Bayesian Structure learning.

4.3.1 Conceptualisation of the problem

To conceptualise the problem at hand, consider a domain whose knowledge can be represented by a Bayesian Network (BN) Structure with four (4) nodes shown in Figure 4.1. Figure 4.1 models the knowledge about acceptance of a research paper for publication. Suppose at time t, it was believed that the acceptance ($acpted(p, j)$) of a paper (p) for publication in journal j depends on whether the paper is a good paper ($GP(p)$), the reputation of the author(s) $RA(p)$ and whether the journal uses double–blind review ($DBR(j)$). This is represented by BN structure in Figure 4.1. This translates to the following First Order Logic (FOL) statements; $\forall p \forall j RA(p) \rightarrow Acpted(p, j)$, $\forall p \forall j GP(p) \rightarrow Acpted(p, j)$, and $\forall p \forall j DBR(j) \rightarrow Acpted(p, j)$ with logical variables p and j representing a research paper and a journal respectively.

Now, suppose that at time $t + 1$, the data emitted from the domain seems not to support the proposition that the acceptance of a paper in a journal depends on the reputation of the authors. The dependency $\forall p \forall j RA(p) \rightarrow Acpted(p, j)$ in Figure 4.2 could have been a result of the fact that reputable authors write good quality papers and as result their papers are accepted on the merit of their quality regardless of the reputation of the authors. In this case the evidence suggests the proposition, $\forall p \forall j RA(p) \rightarrow Acpted(p, j)$, must be given up and the proposition,

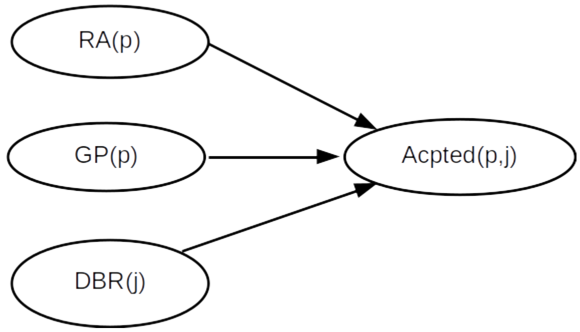


Figure 4.1.: An Example BN structure representing beliefs about papers and their acceptance in Journals.

$\forall p RA(p) \rightarrow GP(p)$, must be accepted. The resulting BN structure can be any one the structures that accepts the edge $\forall p RA(p) \rightarrow GP(p)$. That is, it can be any one of the structures in Figure 4.2. Owing to the fact that network structures are not directly observable, one cannot be certain which of these BN structures is the correct one. One is also not certain as to whether this change is a correction of the incorrect beliefs about the domain held at time t or evidence of a change that has occurred in the domain after time t. If it is evidence of a change in the domain, then there is a possibility that what has been observed is the least of what could have changed in the domain. How then can the beliefs on the dynamics of the knowledge about the domain be captured ensuring that the principle of information economy is adhered to? That is, no beliefs are given up beyond necessity (Rott, 2000). Moreover, one cannot also be sure of when the evidence is compelling enough to effect a change in the beliefs about the domain.

Belief Change tries to address these challenges by giving a mechanism by which the beliefs about the domain should be changed in light of the observations while ensuring minimal change from the beliefs previously held. Typically, Belief Change handles this problem by keeping a plausibility distribution over all the possible BN structures. The most plausible BN structure that supports the evidence is assumed to be the correct BN structure. Belief Revision (Alchourrón, Gardenfors, & Makinson,

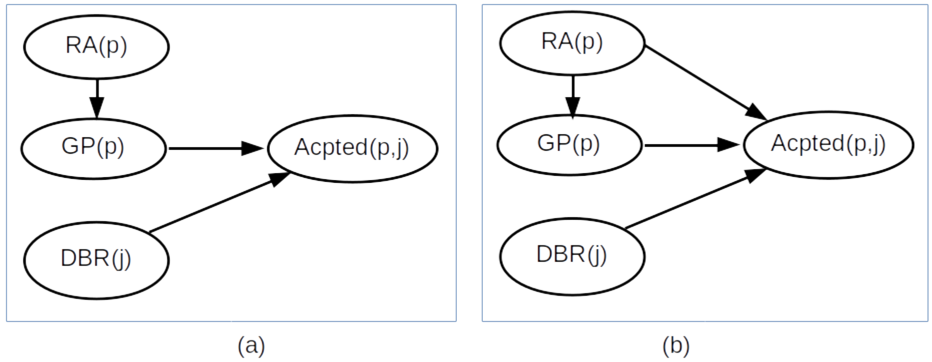


Figure 4.2.: Possible resulting BN Structures after adding the proposition

1985) and Belief Update (Katsuno & Mendelzon, 1991) are the two most widely accepted and debated approaches to Belief Change. Belief Revision focuses on how beliefs should be changed when new information is acquired about a domain that is assumed static. On the other hand, Belief Update focuses on how beliefs should be changed when it is realised that the world has changed. The key difference between these approaches is in how the beliefs should be changed after observing something contrary to what was previously believed. Revision treats the observation of evidence of $\forall p RA(p) \rightarrow GP(p)$ as new information that suggests that the quality of a paper, $GP(p)$, depends on the reputation of the authors(s), $RA(p)$, and accepts the consistent BN structure that is closest to the BN structure accepted at time t that accepts this proposition. On the other hand, Update treats this observation as an indication that a change has occurred in the domain between time t and time $t{+}1$ and the observation of $\forall p RA(p) \rightarrow GP(p)$ is the least that could have resulted from the change in the world. At time t, $GP(p)$ was not dependent on $RA(p)$, but there was a change in the domain that led to this proposition being accepted at time $t+1$. It then tries to find the most likely change that provides an explanation for this observation. This change must result in the BN structure that is closest to the BN structure at time t, among those that accept the new proposition. In real life situations, one would want to consider quite a number of explanations for the observation at time $t{+}1$ depending on what is most likely to have happened. Given a dynamic domain it will be very naïve to assume that the representation at time t was wrong or to assume that the world has changed without a viable explanation. The Unified Belief Change Model by Boutilier 1998 is one of the solutions to this problem in classical Belief Change. The Unified Belief Change Model considers both Update and Revision of a Belief Base given some observations.

Although Belief Change has been extensively studied, few works address the problem of Belief Revision and Belief Update of the network structures of Probabilistic Graphical models (Benferhat, 2010). Only simple forms of revision are considered by propagation algorithms. These forms of Belief Change can be interpreted as accumulation of evidence consistent with the graphical network and reasoning from the evidence using the network structure as the background knowledge (Dubois, 2008). The Network structure itself is never revised nor updated. The evidence is not considered as a new piece of knowledge that is to be integrated in to the BN structure.

This thesis investigates how such Belief Change can be done in Bayesian Network structures. It presents a unified Belief Change operator for Bayesian Network based Knowledge Representations that adheres to the principle of minimal change and caters for both Belief Update and Belief Revision in evolution of Bayesian Network structure.

4.3.2 Structure Learning in Bayesian Networks

A Bayesian network is a pair (B^s, B^p), where B^s, is a Bayesian network structure that encodes the assertion of conditional independence and B^p is a set of probability distributions corresponding to the Bayesian network structure. B^s is a Directed Acyclic Graph (See Figure 4.1) and a node is a domain random variable. Each node/variable X_i in the network structure has a corresponding set (which may be empty) of parent variables, denoted by π_i. X_i is independent of its non-children given its parents. Each arc in the graph represents probabilistic dependencies. The probability density function for the variables in the structure is given by the following equation,

$$P(X_1, X_2, X_3, ..., X_n) = \prod_{i=1}^{n} P(X_i|\pi_i) \tag{4.1}$$

Learning Bayesian network entails estimating the network structure, B^s, from data and then subsequently learning the conditional probabilities, B^p. This paper focuses on learning the Bayesian Network (BN) structure.

Structure learning methods can broadly be categorised into two (2) categories; Constraint-Based, and Search-and-Score methods. Constraint-Based methods are based on Conditional independence tests. The intuition is, if all conditional independencies between variables can be discovered, they can be can be used to construct a Bayesian network. These algorithms use a series of conditional hypothesis tests to learn independence constraints on the structure of the model. Constraint-Based approach algorithms are relatively faster and less computationally demanding compared to Search-and-Score methods when the number of variables is very large, have a well-defined stopping criterion (Dash & Druzdzel, 2003), and are generally asymptotically correct (Cheng et al., 2002; Peter Spirtes & Scheines, 2000). However, some of the assumptions on which these algorithms are based mean that there are certain important classes of association that the algorithms simply cannot detect. These algorithms work on the principle of discarding an edge whenever a conditional independence test fails to rule out independence. They are also unstable owing to their dependence on the threshold selected for conditional independence testing (Dash & Druzdzel, 2003). This renders them unreliable in performing conditional independence tests using large condition sets and a limited data size (Cooper & Herskovits, 1992; Heckerman, Meek, & Cooper, 2006; Peter Spirtes & Scheines, 2000). Even worse, a conditional independence test error can result in a sequence of propagated errors in the subsequent learning process, resulting in an erroneous graph structure (Dash & Druzdzel, 2003; Heckerman, Meek, & Cooper, 2006; Peter Spirtes & Scheines, 2000; Zhao et al., 2015).

Apart from the above-mentioned challenges with constraint-based algorithms, they will also be found wanting if they are to be used for Belief Change in knowledge representations, owing to their lack of a natural way of incorporating a prior network structure in the structure learning process.

Search-and-Score methods (Chickering, 2003; Cooper & Herskovits, 1992; Heckerman, 2008; Heckerman, Geiger, & Chickering, 1995) combine a strategy for searching through the space of possible structures with a scoring function measuring the fitness of each structure to the data. Search-and-Score methods omit the step of removing an edge whenever a conditional independence test fails to rule out independence and proceed directly to evaluating all tentative graph structures provided by some method via a suitable scoring metric. This helps the method to not be too restrictive. However, learning the structure from data by considering all possible structures exhaustively is usually not feasible in most domains (Chickering, 2003), since the number of possible structures grows exponentially with the number of nodes (Cooper & Herskovits, 1992). Hence, structure learning requires either sub-optimal heuristic search algorithms or algorithms that are optimal under certain assumptions. As a result of the foregoing, Search-and-Score algorithms are heuristic and usually have no proof of correctness (Cheng et al., 2002).

Assuming the initial network structure is known, Search-and-Score methods are good candidates for learning evolving network structures for Knowledge representations. This is owing to the fact that they allow incorporation of prior beliefs about the network structure (Heckerman, Geiger, & Chickering, 1995) into the structure learning process. Further to this fact, the learning process is based on conditionalisation which is known to adhere to the principle of minimal change required in Belief Change (Harper, 1975; P. M. Williams, 1980). This makes Search-and-Score methods better candidates for evolving Bayesian network structures for knowledge representation compared to constraint-based methods, though at the cost of efficiency.

Search-and-Score methods search through the space of all possible structures looking for a structure that best fits the data. Such a structure will have the highest probability given the data. The probability of a given network structure given some observations is given by the following equation:

$$P(B^s|D) = \frac{P(D|B^s)P(B^s)}{P(D)} \tag{4.2}$$

Since $P(D)$ is constant for all network structures, it can be taken as a normalising constant. Thus, $P(B^s|D) \propto P(D|B^s)P(B^s) = P(D, B^s)$. This means the network structure that best fits the data has the highest $P(D, B^s)$. To compute $P(D, B^s)$, the following assumptions (Cooper & Herskovits, 1992; Heckerman, Geiger, & Chickering, 1995; J. D. Williams & Young, 2007) are made:

i. The data is observable and a multinomial sample from some Bayesian Network;

ii. The cases appear independently given a Bayesian Network;

iii. There are no cases with variables with missing values;

iv. Parameters associated with different nodes are globally and locally independent, i.e.: (a) the conditional probabilities associated with different nodes are independent of each other, i.e. $P(X_i|\pi_i = \omega_k)$ and $P(X_i|\pi_i = \omega_l)$, $k \neq l$, are independent; (b) the conditional probabilities associated with parents' different instantiations are also independent, i.e. $P(X_i|\pi_i)$ and $P(X_j|\pi_j)$, $i \neq j$, are independent. Here ω_k is the k^{th} value of X_i's parents;

v. Parameters satisfy "Parameter Modularity". If X_i has the same parents in any two Bayesian Networks structures B_1^s and B_2^s, then the parameters with respect to X_i are the same for both structures;

vi. The prior knowledge about the possible BN structure can be expressed as the prior probability distribution over all possible BN structures, $P(B^s)$.

Theorem 4.1: Let $\mathbf{X}$ be a set of n discrete variables, where a variable in $\mathbf{X}$ has r_i possible value assignments: $(v_{i1}, v_{i2}...,v_{ir})$. Let D be a database of m cases, where each case contains a value assignment for each variable in X. Let B^s denote a belief-network structure containing just the variables in $\mathbf{X}$. Each variable X_i in B^s has a set of parents, which we represent with a list of variables . Let w_{ij} denote the j^th unique instantiation of π_i relative to D. Suppose there are q_i such unique instantiations of π_i . Define N_{ijk} to be the number of cases in D in which variable X_i has the value v_{ik} and π_i is instantiated as w_{ik} . Let $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$. Given the foregoing assumptions;

$$P(D, B^s) = P(B^s) \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(N'_{ij})}{\Gamma(N'_{ij} + N_{ij})} \prod_{k=1}^{r_i} \frac{\Gamma(N'_{ijk} + N_{ijk})}{\Gamma(N'_{ijk})} \tag{4.3}$$

This is what is generally known as the Bayesian Dirichlet (DB) score. The DB score can be extended to the Bayesian Dirichlet equivalence BDe score (Heckerman, Geiger, & Chickering, 1995), which is the most popularly used score metric.

4.4 Formalising the Logical Bonds between Belief change and change in BN Structure

This section discusses the formalisation of the Belief Change problem on Bayesian Network Structure in terms of the logical structures that are used in Belief Change. The discussion will start off by discussing how beliefs will be represented, and then give the postulates for a Unified Belief Change Model for rational Belief Change in Structure in Bayesian Networks.

In the AGM (Alchourrón, Gärdenfors, & Makinson, 1985) and the KM (Katsuno & Mendelzon, 1991), beliefs are represented by a set of logical sentences. In the AGM theory, beliefs are represented by a Belief Set, and in the KM theory by a Belief Base. A Belief Set is a set of logical sentences assumed to be closed under logical consequence, whereas a Belief Base is a set of logical sentences that is not (except as a limiting case) closed under logical consequence (Hansson, 2017). The distinction on whether to use a Belief Set or Belief Base as a representations of beliefs emanates from one of the most debated topics in Belief Revision, the recovery postulate (Hansson, 2017). The recovery postulates states that all the original beliefs are regained if one of them is first removed and then reinserted (Makinson, 1987). The recovery postulate holds in Belief Change models based on Belief Sets, and it does not hold in those based on Belief Bases. Whilst for Belief Sets two equivalent Belief Sets may behave differently under operations of change, two equivalent Belief Bases always behave the same under operations of change (Hansson, 2017). As a result, it is at times preferred to work with Belief Bases to gain dynamic equivalence between two equivalent Belief Bases at the expense of the recovery. In this work because, we are proposing a Belief Change solution that caters for both Belief Revision and Belief Update, we use Belief Bases rather than Belief Sets for representation of Beliefs.

In classical logic, propositions are viewed as having a certain set-theoretic structure. This view enables representation of propositions as a set of Possible Worlds in which the propositions are true. This section discusses how such a set theoretic structure can be defined over a set of all possible Bayesian Network structures as a set of possible worlds. We take propositions to be objects of belief. A proposition is represented by a set of possible BN structures (possible worlds) for which the proposition is true. Subjective probabilities will be used as the degree of belief. This thesis defines an Algebra, $\mathcal{A}$, over a set of all possible BN structures $\mathcal{W}$ as follows:

Definition 4.1 : *$\mathcal{A}$ is an algebra over $\mathcal{W}$ iff $\mathcal{A} \subseteq \mathcal{P}(\mathcal{W})$, the power set of $\mathcal{W}$, such that given proposition A and B (Spohn, 2012),*

a. $\mathcal{W} \in \mathcal{A}$

b. if $A \in \mathcal{A}$, then $\overline{A} \in \mathcal{A}$

c. if $A, B \in \mathcal{A}$, then $A \cup B \in \mathcal{A}$

d. for each countable $\mathcal{B} \subseteq \mathcal{A}$, $\bigcup \mathcal{B} \in \mathcal{A}$. That is $\mathcal{A}$ is a sigma algebra.

A and B are propositions represented using the possible world model view, such that $A = \{B_i^s : B_i^s \models A\}$ and $B = \{B_i^s : B_i^s \models B\}$.

Let $K = \{\alpha_{ji} :$ there is an edge from node X_j to X_i, $j \neq i\}$ be a set of propositions about the existence of a dependence between any two random variables characterising the domain. In classical logic such a set is known as a Belief Base. Any BN structure,

B_i^s, that satisfies α_{ji} (that is, $B_i^s \models a_{ji}$ or the dependence (edge) α_{ji} exists in B_i^s) is dubbed the $\alpha_{ji} - world$. Henceforth, the symbols, A, and the α_{ji} will be used interchangeably to denote a proposition. The only difference is, the symbol α_{ji} will be used to explicitly represent a proposition as a sentence, and A to represent the proposition as a set of worlds (BN Structures) that entails the sentence α_{ji}.

Using the afore-defined algebra, a probability function (p-function) over the subset of the Belief Base K is defined as normalised weighting function $P : \mathcal{W} \to [0, 1]$ that satisfies the classical Kolmogorov axioms of probability:

1. $0 \leq P(A) \leq 1$
2. $P(K) = 1$
3. $P(\bigcup A_i) = \sum P(A_i)$, when $A_i \cap A_j = \emptyset$

Drawing inspiration from (Boutilier, 1995), rather than taking Belief Bases as primitive, the researcher postulates that the primitive component of the epistemic state is the p-function from which the Belief Base K can be derived. Definition 4.2 therefore follows from this position.

Definition 4.2:*A p-function over the possible BN structures, P, is compatible with a Belief Base K just when $P(A) = 1$ iff $A \in K$ (Boutilier, 1995)*

Such a *p-function* over the possible BN structures induces a *p-function* over the proposition via the standard relationship;

$$P(\alpha_{ji}) = \sum_{B_i^s \models \alpha_{ji}} P(B_i^s) \tag{4.4}$$

where $B_i^s \models \alpha_{ji}$ means B_i^s semantically entails α_{ji}.

The Belief Base at any given time is the top of the *p-function* over the propositions (i.e. those α_{ij} such that $P(\alpha_{ji}) = 1$. Thus, the Belief Base K is a set all the edges that exist in all the BN structues in $\mathcal{W}$.

Definition 4.3: *The Belief Base induced by a p-function is given by top of a p-function, i.e. $K = \{\alpha_{ji} : P(\alpha_{ji}) = 1\}$.*

The top of the p-function can be relaxed to be bounded below by a value less than one (1) without loss of generality. For instance the top of the p-function can be relaxed to be bounded below by 0.95 such that the Belief Base K is induced from the p-function as follows; $K = \{\alpha_{ji} : P_t(\alpha_{ji}) \geq 0.95\}$

A rational Belief Change operator should therefore revise and/or update the Belief Base K with α_{ji} such that the principle of information economy is adhered to. This work is therefore an effort to define a Belief Change Model for structure change in Bayesian Networks that builds on the formal bonds that the structure change problem has with the Belief Change problem in classical logic. In the next subsection, we define the postulates that such a model should satisfy.

4.4.1 Postulates for Unified Belief Change in Bayesian Network Structures

This thesis postulates that the p-function over all possible BN structures holds all the information needed for rational Belief Change in the BN structure, and the Belief Base induced by the p-function captures the current beliefs about the world. The goal in this section is therefore to formulate the postulate for rational Belief Change of structure in Bayesian networks. The underlying principle for rational Belief Change is that the Belief Change operator should return as much as possible from the old beliefs.

This is generally referred to as the principle of *informational economy* or *minimal change*. This Thesis assumes that for every p-function, P, and its corresponding Belief Base, K, there is a unique p-function, $P_A^{\triangleleft}$ and a corresponding Belief Base $K_A^{\triangleleft}$ representing the results of rational Belief Change on the Belief State by A using a given Belief Change operator, $\triangleleft$.

Boutilier (1995) modified the postulates for revision of p-functions proposed by Gardenfors (1988) to give a modified set of postulates that a probabilistic revision function should satisfy. We further modified the postulates in order for them not to be invalid for a unified Belief Change Model. Below is a set of Belief Revision postulates we defined for the Unified Belief Change Model:

(P1) $P_A^{\triangleleft}$ is a consistent p-function

(P2) $P_A^{\triangleleft}(A) = 1$

(P3) If the domain is static and $P(A) > 0$, then $P_A^{\triangleleft} = P(.|A)$

(P4) If $P_A^{\triangleleft}(B) > 0$, then $P_{A \wedge B}^{\triangleleft} = P_A^{\triangleleft}(.|B)$

Postulate $P1$, requires that an output of a Belief Change on a consistent *p-function* by a proposition should also be a p-function. This satisfies the principle of consistency maintenance. $P2$ guarantees that if the p-function is revised with respect to proposition A, the Belief Base induced by the *p-function* should accept A (i.e A should be in the top of the resulting *p-function*). This is generally known as the success postulate. $P3$ ensures that if the domain is static no update will be effected by the operator. Thus, the Belief Change operator will only effect revision.

4.5 Deriving the Unified Model for Belief Change in Bayesian Networks

The problem of Belief Change in Bayesian Network Structure is in this thesis presented as dynamic process that can be modelled using a Dynamic Bayesian Network. This is not a far-fetched hypothesis, since Bayesian modelling have over the years been used as a mechanism for Bayesian Network Structure Learning (Chickering, Heckerman, & Meek, 1997; Eggeling et al., 2019; Friedman & Koller, 2003; Han et al., 2017). The hypothesised dynamic Bayesian Network is taken to be the Belief Change Meta-Model that will be used to conceptualise the Belief Change Model developed in this thesis.

At any given point in time, the actual true network structure is unobservable, hence it is not known with certainty, so a distribution over all the possible structures is maintained. Such a distribution is known as the Belief State in Belief Change literature (Boutilier, 1995). This thesis uses $P_t(B^s)$ to denote such a distribution at a given time t. The Belief Base, or set of sentences accepted is an abstraction over the p-function given by the top of the p-function (Boutilier, 1995).

Definition 4.4: *The Belief Base induced by a p-function is given by top of a p-function, i.e.* $K = \{A : P_t(A) = 1\}$*, where* A *is some proposition*

According to Gardenfors (1988), Lindstrom and Rabinowicz (1989), and Boutilier (1995) non-beliefs can be further discriminated to give different degrees to each one of them. The set of non-beliefs consists of all $A : 0 < P(A) < 1$. These are necessary because they hold the information necessary for the Belief Change Operator. Boutilier (1998) argued that system dynamics can be characterised by two families of probability distributions. These are event probabilities and outcome probabilities. Event probabilities, $P_t(e|B^s)$, model the likelihood of a given event,e , occurring given that B_s is the actual state (network structure) of the world. The outcome probabilities, $P_t(B^{s\prime}|B^s, e)$, model the probability of a state (network structure), resulting from occurrence of event,e , in state B^s. The unrolled Dynamic Bayesian Network in Figure 4.3 models the regularities we postulated for structure change in Bayesian networks that defines the proposed Belief Change Meta-Model. The event vocabulary for BN structure change abstractly consists of the following event classes: (i) addition of an edge; (ii) deletion of an edge, and (iii) reversal of an edge.

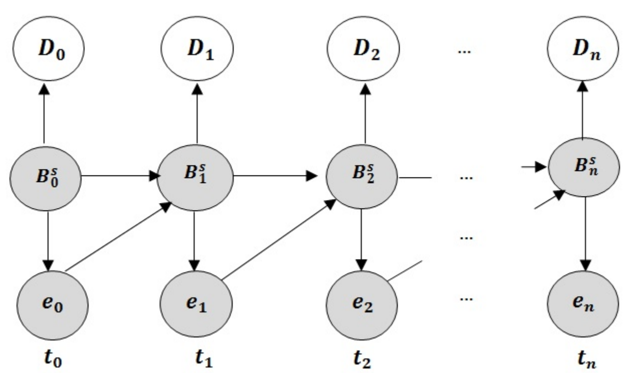


Figure 4.3.: A Dynamic Bayesian Network for Modelling Structure evolution in a Bayesian Network

The B_t^ss in the unrolled Dynamic Bayesian network in Figure 4.3 represent the actual states (network structures) at a given time t. The e_ts represent events that can possibly occur given that B_t^s is the true state at time t. D_t represents the data emitted from the true state/structure at time t. The events in this case will be removal of an edge, addition of an edge, and reversal of an edge. In the DBN in Figure 4.3, both the states and the events are unobservable. This is owing to the fact that in learning Probabilistic graphical models, we cannot directly observe addition, deletion, or reversal of edges. These events can only be inferred based on the observable data, D^t, emitted from the current structure, B_t^s.

On the basis of DBN in Figure 4.3 and the definition of the Unified Belief Change model (Boutilier, 1998) for qualitative Belief Change, this research defines an event-based model for rational Belief Change of the structure of a Bayesian Network. Events are defined in terms of the three (3) operators for introducing local variations to an existing Bayesian Network, insertion of an edge, removal of an edge, and reversal of an edge, and the null event. An event is defined as follows:

Definition 4.5: *An event, e, is a possible local variation of a network structure that maps each structure into a probability distribution over possible BN structures, $e : B^s \rightarrow (B^s \rightarrow P)$. We use $P(B^{s\prime}|B^s, e)$ to denote this distribution.*

Since these events are unobservable and there is no certainty on which event caused the observation, an event probability distribution associated with each network structure, is defined as follows;

Definition 4.6: *An event probability function, μ, maps each Network Structure into a probability distribution over events, $\mu : B^s \rightarrow (E \rightarrow P)$, where $E \rightarrow P$ captures the probability distribution of events likely to occur in Belief State. $P(e|B^s)$ is used to denote the event probability distribution defined for a given network structure B^s.*

For sure events, $P(e|B^s) = 1$ and $P(e|B^s) = 0$ for impossible events. Finally we define the Unified Belief Change model for rational Belief Change of structure in Bayesian Networks as follows;

Definition 4.7: *A Unified Belief Change Model for Bayesian Network structures has the form $M = \langle \mathcal{W}, P, E, \mu, \lhd \rangle$; where $\mathcal{W}$ is the set of possible Network structures, P is a probability distribution (Belief State) over possible Bayesian Network structures, E is a set of events (local variation) that can occur in each Bayesian Network structure, μ is a set of probability distribution over events that can occur given Bayesian Network structure and $\lhd$ is the Belief Change Operator.*

Definition 4.6 defines an Epistemic Space over which the Epistemic State is defined. The Epistemic State consists of two groups of distributions, the Belief State,P , and the event distributions ,μ. Belief Change is effected on these two sets of distributions and the result of the Belief Change is also set of these distributions.

Boutilier (1998) postulated that a Unified Belief Change Model proceeds in two phases. First, the agent updates its Belief State, and second it revises this Belief State by the observation. In the following two subsections, we are going to discuss the conceptualisation of these two phases for Belief Change in Bayesian Networks.

4.5.1 Updating the Belief State

Now let's assume that the system dynamics have moved one timestep forward from time t to $t+1$. Assuming the dependencies modelled in Figure 4.3 and making a Markov assumption of order 1, the joint probability distribution at time $t+1$ over the random variable $B^{s\prime}$, representing the state (network structure) at time $t+1$, the state at time t, and e the event that can occur at time t, is given by Equation (4.5);

$$P_{t+1}(B^{s\prime}, B^s, e) = P_t(B^{s\prime}|B^s, e)P_t(e|B^s)P_t(B^s) \tag{4.5}$$

The Belief State (or probability distribution over states B^s) at time $t+1$ is the marginal probability of $B^{s\prime}$ obtained by marginalising out the latent variables B^s and e, giving Equation (4.6).

$$P_{t+1}(B^{s\prime}) = \sum_{B^s \in \mathcal{W}} \sum_{e \in E^s} P_t(B^{s\prime}|B^s, e)P_t(e|B^s)P_t(B^s) \tag{4.6}$$

where $\mathcal{W}$ is a set of all possible states/network structures, E^s is the set of all events that can occur in state B^s.

Equation (4.6) can also be derived either by analogy from Boutilier's qualitative Unified Belief Change Model or from the theory on Partially Observable Markov Decision Processes (POMDPs). The next two Subsections are going to derive Equation (4.6) from both the above mentioned perspectives.

However, a closer look at Equation (4.6) reveals that there is only one event that can occur in state B^s resulting in state $B^{s\prime}$. This implies that $P_t(B^{s\prime}|B^s, e) = 0$ for all e that does not result in $B^{s\prime}$ and $P_t(B^{s\prime}|B^s, e) = 1$ otherwise. If we define the null event as the event that no edge insertion, removal or reversal happens and as a result the network structure does not change (i.e. $B^{s\prime} = B^s$), Equation (4.6) can be rewritten as follows;

$$P_{t+1}(B^{s\prime}) = \sum_{B_s \in \mathcal{W}} \sum_{e \in E^s} I(B^{s\prime}|B^s, e) P_t(e|B^s) P_t(B^s) \tag{4.7}$$

where E^s is the set of all events that can occur in state B^s, including the null event, and $I(B^{s\prime}|B^s, e)$ is an indicator function which is equal to one(1), if occurrence of event e in state B^s results in state $B^{s\prime}$ and zero (0) otherwise. Equation (4.7) is important for efficient implementation of the update operator. Computation of some parameters for events that do not result in legal states can be skipped.

4.5.1.1 Validating the Update Model by Analogy to the Unified Qualitative Update Model

The unified Belief Change Model as proposed by Boutilier (1995), assumes that system dynamics are governed by two families of probability functions. These are event probabilities and outcome probabilities. Events are assumed to provide an impetus for change and the plausibility of a Belief Change is determined by the plausibility of the events that can cause the change.

An event e, maps each world into a partial ranking over worlds, $e : W \rightarrow (W \rightarrow N)$, where N is a set of natural numbers. This ranking gives a qualitative probability distribution over worlds, which is denoted by κ_{we}. Intuitively $\kappa_{w,e}(v)$ describes the plausibility that the world v results when event e occurs at world w. The most likely outcomes are given the rank zero (0). v is an impossible outcome of event e occurring in world w if $\kappa_{w,e}(v) = \infty$. The expression $\kappa_{w,e}(v)$ should be equal to zero for at least one event. A null event is defined to cater for static domains. In static domains, for a null event, n, $\kappa_{wn}(v) = 0$ for $w = v$ and $\kappa_{we}(v) = \infty$ for any $w \neq v$. A null event thus ensures that the world does not change. For any given world an event ordering to capture the probability distribution of events is defined as follows.

An event ordering μ maps each world into a partial ranking over events, E, $\mu : W \rightarrow (E \rightarrow N)$. This gives a qualitative pausability distribution of possible events in a given world denoted by κ_w. $\kappa_w(e)$ captures the plausibility of the occurrence of an event e in world w. An event e is impossible if $\kappa_w(e) = \infty$.

The Belief State of the system is reflected in the ranking of the worlds (states) κ. $\kappa(w)$ denotes the qualitative plausibility distribution over all possible states. $\kappa(w) = \infty$ means the world w is impossible.

The generalized update model is defined as a tuple,$M = (W, \kappa, E, \mu)$, where W is a set of possible worlds, κ is a ranking (plausibility distribution) over worlds (κ is also known as the Belief State of the system), E is a set of possible events defined over given states, and μ is the event ordering (plausibility distribution of events over given worlds).

Boutilier (1995) postulated that the plausibility of a transition from w to v after occurrence of event e depends on the plausibility of w, the likelihood that event e occurred and the likelihood of the outcome v given event e occurred in state w; i.e

$$\kappa(w \xrightarrow{e} v) = \kappa_{w,e}(v) + \kappa_w(e) + \kappa(w) \tag{4.8}$$

The updated ranking (qualitative probability distribution), $\kappa^{\diamond}$ is given by (Boutilier, 1998):

$$\kappa^{\diamond}(v) = \min_{w \in W, e \in E} \kappa_{w,e}(v) + \kappa_w(e) + \kappa(w) \tag{4.9}$$

Addition in qualitative probability theory is equivalent to multiplication in quantitative probability theory and minimisation is equivalent to addition in quantitative probability distributions (Friedman and Halpen, 1996). This implies that $\min_{w \in W, e \in E}$, is equivalent to $\sum_{s \in S}, \sum_{e \in E}$ in quantitative probability theory, and after replacing the ranks with probabilities, the quantitative analogy of the expression $\kappa_{w,e}(v) + \kappa_w(e) + \kappa(w)$ is given by, $\sum_{w \in W} \sum_{e \in E} P(v|w,e)P(e|w)P(w)$. As a result the equivalent of Equation (4.9) in quantitative probability theory is $\sum_{w \in W} \sum_{e \in E} P(v|w,e)P(e|w)P(w)$ with the ranks replaced with probabilities. This is the updated marginal distribution of the states analogous to Equation (4.6).

4.5.1.2 Validating the Update model from the Theory on POMDP

A Partially Observable Markov Decision Processes (POMDP) models an agent's decision process in which it is assumed that the system dynamics are determined by a Markov Decision Process (MDP). The agent performs some actions that can change the state of the world, but it cannot directly observe the underlying state. The agent only maintains a probability distribution over all possible states, called a Belief State and a mechanism/operator for updating the Belief State on the basis of the observations emitted from the latent state.

Formally a POMDP as a tuple $\{S, A, Tr, R, \Omega, O\}$ (Shani, 2007), where;

- S, is a set of possible states,
- A, is a set of actions that an agent can take,
- Tr, defines, a transition probability $P(s'|s,a)$. s' is the resulting state after an agent perform action a in state s.
- R, defines the expected reward for performing action a in state s,
- Ω, is a set of observations

- O, is the probability that an agent will observe after executing a, reaching state s',$p(o'|s', a)$

At each timestep the world is in some unobserved state s. The agent selects an action a, and the world transitions to an unobserved state s'. This transition only depends on the state s and the action a. Convention in POMDP defines $b(s)$ to be probability distribution of being in a particular state s. At each time stamp the Belief State is updated to a new Belief State $b'(s')$ on the basis of the evidence as follows (J. D. Williams & Young, 2007):

$$\begin{aligned} b'(s') &= P(s'|o', a, b) \\ &= \frac{P(o'|s', a, b)P(s'|a, b)}{P(o'|a, b)} \\ &= \frac{P(o'|s', a)\sum_{s \in S} P(s'|a, b, s)P(s|a, b)}{P(o'|a, b)} \\ &= \frac{P(o'|s', a)\sum_{s \in S} P(s'|a, s)b(s)}{P(o'|a, b)} \end{aligned} \tag{4.10}$$

The numerator of Equation (4.10) is the product of the observation function and the probability distribution of being in a given state after moving one timestep forward. The denominator is independent of s, and can be regarded as the normalisation constant.

In the Belief Change process for Bayesian Network structures modelled in Figure 3, unobservable events (instead of observable agent actions in classical POMDP) that result from the system dynamics provide an impetus for change. Agents can only make epistemic actions in order revise and update their beliefs about the domain. The function, $\sum_{s \in S} P(s'|a, s)b(s)$ is the one that deals with Belief Update and it gives the probability distribution of being in a given state after moving one timestep forward. Replacing the agent's actions with events, this function results in Equation (4.11):

$$\sum_{s \in S} P(s'|e, s)b(s) \tag{4.11}$$

However, events, unlike actions in the case of POMDP are unobservable. Since e, is a hidden/latent variable, we can replace e by its distribution in Equation (4.11) and marginalise it out as follows:

$$\sum_{s \in S} P(s'|e,s)b(s) = \sum_{s \in S} \sum_{e \in E} P(s'|e,s)P(s)P(e|s) \tag{4.12}$$

Since $b(s)$ is the probability distribution over possible states/worlds at the previous timestep, it can be replaced by $P(B^s)$. To capture the concept of change in BN structure, $P(s'|e,s)$ is replaced with $P(B^{s'}|B', e)$, and $P(e|s)$ with $P(e|B^s)$. Equation (4.12) can then be restated as follows:

$$\sum_{B^s \in \mathcal{W}} P(B^{s'}|B^s, e)P(B^s) = \sum_{B^s \in \mathcal{W}} \sum_{e \in E} P(B^{s'}|B^s, e)P(e|B^s)P(B^s) \tag{4.13}$$

Equation (4.13) gives the probability distribution of being in a given state after moving one timestep forward and is the same as Equation (4.6).

4.5.2 Revising the Belief State

The second phase of the Belief Change process in dynamic domains should then cater for revision of the updated Belief State based on the evidence D. This is done by providing the best possible explanation for what has been observed. In the qualitative Belief Change Model, this is done by finding high ranking states among those that are supported by the evidence and can possibly have resulted from some event occurring in some likely world/state. This can be interpreted to mean that the revised Belief State should give higher rankings to states that are supported by the evidence D. States that are not supported by data, even though they may have ranked high after update they are not considered. This is in principle conditionalisation, and from a quantitative perspective the resulting Belief State after revision is given by the following equation:

$$\begin{aligned} P^{D}_{t+1}(B^{s\prime}) = P_{t+1}(B^{s\prime}|D) &= \frac{P_{t+1}(D|B^{s\prime})P_{t+1}(B)}{P_{t+1}(D)} \\ &= \frac{P_{t+1}(D|B^{s\prime})P_{t+1}(B^{s\prime})}{\sum_{B^{s\prime}\in\mathcal{W}} P(D|B^{s\prime})P_{t+1}(B^{s\prime})} \\ &= \frac{P_{t+1}(D|B^{s\prime})\sum_{B^{s}\in\mathcal{W}}\sum_{e\in E^{s}} P_t(B^{s\prime}|B^{s},e)P(e|B^{s})P_t(B^{s})}{\sum_{B^{s\prime}\in\mathcal{W}} P_{t+1}(D|B^{s\prime})\sum_{B^{s}\in\mathcal{W}}\sum_{e\in E^{s}} P_t(B^{s\prime}|B^{s},e)P(e|B^{s})P_t(B^{s})} \end{aligned} \tag{4.14}$$

Equation (4.14) is equivalent to Equation (4.10) except for the fact that in Equation(4.14) observations are independent of the events given the hidden state and that events are unobservable (see the DBN in Figure 4.1).

4.6 The Unified Belief Change Model and the Revision function Postulates

This Section presents the substantiation that the proposed Unified Belief Change Model satisfies the postulates discussed in Section 4.3. The postulates are listed below for easy reference:

(P1) If A is satisfiable, then $P^{\triangleleft}_{A}$ is a (consistent) p-function.

(P2) If A is satisfiable, then $P^{\triangleleft}_{A}(A) = 1$

(P3) If the domain is static and $P(A) > 0$, then $P^{\triangleleft}_{A} = P(.|A)$

(P4) If $P^{\triangleleft}_{A}(B) > 0$, then $P^{\triangleleft}_{A\wedge B} = P^{\triangleleft}_{A}(.|B)$

Proposition 4.1: *If A is satisfiable, then $P^{\triangleleft}_{A}$ is a (consistent) p-function*

Proof of Proposition 4.1: This follows from the derivation of the p-function resulting from our Belief Change Model. The resulting function is a consistent p-function.

Proposition 4.2: *if A is satisfiable then $P^{\triangleleft}_{A}(A) = 1$*

Proof of Proposition 4.2: Let $P' = P_{t+1}(.|A)$ be the new distribution after Belief Change by A. If A is satisfiable, then $P'(A|B^{s}_{i}) = 1$, since if A is satisfiable, it implies that $A \in B^{s}_{i}$, $\forall B^{s}_{i} \in \mathcal{W}$:

$$P'(A|B^{s}_{i}) = \frac{P'(B^{s}_{i}|A)P'(A)}{P'(B^{s}_{i})}$$

After rearranging the terms, $P'(A)$ is given by the following equation,

$$P'(A) = \frac{P'(A|B_i^s)P'(B_i^s)}{P'(B_i^s|A)}$$

But $P'(A|B_i^s) = 1$. This gives

$$P'(A) = \frac{P'(B_i^s)}{P'(B_i^s|A)}$$

$$P'(A)P'(B_i^s|A) = P'(B_i^s)$$

Taking summations of both sides

$$\sum_{B_i^s \in \mathcal{W}} P'(A)P'(B_i^s|A) = \sum_{B_i^s \in \mathcal{W}} P'(B_i^s)$$

$$P'(A) \sum_{B_i^s \in \mathcal{W}} P'(B_i^s|A) = \sum_{B_i^s \in \mathcal{W}} P'(B_i^s)$$

$$P'(A) = \frac{\sum_{B_i^s \in \mathcal{W}} P'(B_i^s)}{\sum_{B_i^s \in \mathcal{W}} P'(B_i^s|A)}$$

The terms $\sum_{B_i^s \in \mathcal{W}} P'(B_i^s)$ and $\sum_{B_i^s \in \mathcal{W}} P'(B_i^s|A)$ are both equal to one (1), since they are summing probabilities over the whole sample space. This results in

$$P'(A) = \frac{1}{1} = 1$$

Proposition 4.3: *If* $P(A) > 0$ *and the domain is assumed static, then* $P_A^{\triangleleft} = P(.|A)$.

Proof of Proposition 4.3: This follows from proposition 4.2. If the domain is static, $P_{t+1} = P_t$. This implies that $P_A^{\triangleleft} = P_{t+1}(.|A) = P_t(.|A)$. Thus $P_A^{\triangleleft} = P(.|A)$.

Proposition 4.4: *if* $P_A^{\triangleleft}(B) > 0$ *then* $P_{A \wedge B}^{\triangleleft} = P_A^{\triangleleft}(.|B)$

Proof of Proposition 4.4: The Belief Change Operator starts by updating the Belief State, based on the system dynamics and then revising it by an observation that best explains what could have happened. Thus, assuming the new distribution after update P_{t+1} is P', then $P_a^{\triangleleft}(.|B)$ can be rewritten as $P'(.|(B|A))$.

Supposing C is some proposition,

$$P'(C|(B|A)) = \frac{P'((B|A)|C)P'(C)}{P'(B|A)}$$

After rewritting the above equations as a product of its components the following is obtained,

$$\begin{aligned}
P(C|(B|A)) &= P'((B|A)|C)\frac{1}{P'(B|A)}P'(C) \\
&= \frac{P'(B \wedge A|C)}{P'(A|C)}\frac{P'(A)}{P'(B \wedge A)}P'(C) \\
&= P'(B \wedge A|C)\frac{P'(1)}{P'(A|C)}\frac{P'(A)}{P'(B \wedge A)}P'(C) \\
&= \frac{P'(B \wedge A \wedge C)}{P'(C)}\frac{P'(c)}{P'(A \wedge C)}\frac{P'(A)}{P'(B \wedge A)}P'(C)
\end{aligned}$$

Making a Markov assumption of order 1 based on the Markov model in Figure (4.3) and the distribution $P_A^{\triangleleft}(.|B)$, it is concluded that proposition C is independent of the Belief Change by proposition A given revision by B. This implies that $P'(A \wedge C) = P'(C)P'(A)$. Hence

$$\begin{aligned}
P'(C|(B|A)) &= \frac{P'(A \wedge B \wedge C)}{P'(B \wedge A)} \\
&= \frac{P'(A \wedge B \wedge C)}{P'(B \wedge A)}\frac{P'(C)}{P'(C)} \\
&= \frac{P'((A \wedge B) \wedge C)}{P'(C)}\frac{P'(C)}{P'(B \wedge A)} \\
&= \frac{P'((A \wedge B)|C)}{1}\frac{P'(C)}{P'(B \wedge A)} \\
&= \frac{P'((A \wedge B)|C)P(C)}{P'(B \wedge A)} \\
&= P'(C|A \wedge B)
\end{aligned}$$

The distribution, $P'(.|A \wedge B) = P_{t+1}(.|A \wedge B)$, can be rewritten in our Belief Change notation as $P_{A \wedge B}^{\triangleleft}(.)$

On the basis of the foregoing propositions and their proofs, we therefore conclude that the Belief Change operator defined in this thesis satisfies the postulates of a probabilistic revision function defined in (Boutilier, 1998).

4.7 Discussion and conclusions

This chapter proposed a Belief Change model for Bayesian Networks for knowledge representation in Dynamic domains. The Model was derived based on a Belief Change meta-model that this thesis postulated to be the foundational model for Belief Change in dynamic domains. The Belief change meta-model was informed by the event-based semantics for Belief Change proposed by Boutilier (1998). The derived Unified Belief Change Model was formally validated by analogy using the Qualitative Belief Change Model for Dynamic environments, and theory of Partially Observable Markov Decision Processes (POMDP). The semantics of the derived model were found to be very similar to the semantics of POMDPs, except for the fact that the derived model has unobservable events, instead of ontic agent action, providing an impetus for change in the domain. It was also proven that the proposed Belief Change Model meets the postulates for revision of p-functions.

The general approach to Belief Change in Bayesian Networks is similar to the approaches that have emanated from the work on Sequential Update of Bayesian Networks (which is also referred to as incremental Learning of Bayesian Networks in some literature) (Friedman & Goldszmidt, 1997; Lam & Bacchus, 1994; Yu, 2019; Yue et al., 2015). Amongst these works. the work by Friedman and Goldszmidt 1997 is much closer to the approach to Belief Change in Bayesian Networks proposed in this thesis. This is in the sense that after every structure learning operation, the model keeps a set of high-scoring Bayesian Network candidates, that serves as the epistemic state for the next iteration of Bayesian structure learning. When new data is observed the structure learning process searches within the set of the high-scoring BN structures for a structure that best explains the data. However, although these solutions claim to be addressing changes necessitated by both errors in the initial models and changes in the domain, only cater for Belief Revision when they are analysed from a Belief Change perspective.

5. UNIFIED BELIEF CHANGE OPERATOR FOR BAYESIAN NETWORKS (*UBCOBAN*)

5.1 Introduction

In this chapter, an instance of a belief change operator induced by the model presented in Chapter 4 is defined. This will be followed by an illustration of how the operator works on a synthetic toy example.

5.2 The Unified Belief Change Operator for Bayesian Networks (*UBCOBaN*)

Given the Unified belief change model presented in Chapter 4, quite a number of belief change operators can be induced from it. This section defines one such operator and how the parameters for the operator will be determined.

5.2.1 Definition of the *UBCOBaN* Operator

Proposition 5.1: *A rational belief change operator for Bayesian Network structures first updates the belief state (probability distribution over states) on the basis of the current probability distribution over the possible network structures, event probability distributions and likely transitions, and then revise the updated structure by conditioning on the new evidence observed.*

Proof of Proposition 5.1. The proof follows directly from the conceptualisation of the belief change model in Section 4.4;

$$P^D_{t+1}(B^{s\prime}) = \frac{P_{t+1}(D|B^{s\prime}) \sum_{B^s \in \mathcal{W}} \sum_{e \in E^s} P_t(B^{s\prime}|B^s, e) P(e|B^s) P_t(B^s)}{\sum_{B^{s\prime} \in \mathcal{W}} P_{t+1}(D|B^{s\prime}) P_{t+1}(B^{s\prime})} \tag{5.1}$$

The expression $\sum_{B^s \in \mathcal{W}} \sum_{e \in E^s} P_t(B^{s\prime}|B^s, e)P(e|B^s)P_t(B^s)$, captures all the information required for updating the Bayesian Network structure. This can be computed independent of the observations, D. Hence the belief change algorithm can be designed by first computing this function, which updates the belief state and then revising the updated belief state through conditioning on the evidence using Equation (5.1).

Proposition 5.2: *The operator defined in Proposition 5.1 satisfies both belief update and revision.*

Proof of Proposition 5.2. If the domain is static, the only possible event is the null event, n,the event that nothing happens/changes. This event is sure in a static domain, that is $P(n|B^s) = 1$. The probability of the structure changing from one state to another given a null event has occurred is zero for all $B^{s\prime} = B^s$ and one (1) for all $B^{s\prime} \neq B^s$. Substituting these probabilities in Equation (4.6) gives $P_{t+1}(B^{s\prime}) = P_t(B^s)$.

$P_{t+1}(B^{s\prime}) = P_t(B^s)$ implies that the belief state at time t is the same as the one at time $t + 1$. This ensures that our belief change operator does not effect update in domains that are assumed static. Update will only be effected in dynamic domains.

However, as shown in Equation (4.7) the nature of the belief change operator defined in this paper dictates that $P(B^{s\prime}|B^s, e)$ assigns a probability of one (1) to the only event that results from event e occurring in state and zero to all the other states.

For Belief Revision, the Bayesian based Search-and-Score structure learning approach is used. The DB metric defined in Theorem 4.1 (see Equation (5.2)) or any of its extensions can be used as the scoring metric in searching for the structure that best fits the data. (Heckerman, Geiger, & Chickering, 1995).

$$P(D, B^s) = P(B^s) \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(N'_{ij})}{\Gamma(N'_{ij} + N_{ij})} \prod_{k=1}^{r_i} \frac{\Gamma(N'_{ijk} + N_{ijk})}{\Gamma(N'_{ijk})} \tag{5.2}$$

5.2.2 Epistemic States, Belief Bases, and Computation of Parameters

The number of possible structures is usually very large in practice, making the task of effecting the update operator computationally expensive owing to the fact that it will need computation of structure priors, $P_t(B^s)$, and event probabilities, $P_t(e|B^s)$, for all the possible BN structures. The size of these parameters grows exponentially with the size of the Bayesian Network. Taking inspiration from Madigan and Raftery (1994), we take the standard approach of scientific investigation. In a scientific investigation, models are compared on the basis of how well they predict the observations. Models that predict the observation less well than their competitors are discarded. Against this background, this thesis makes the proposal that all BN structures that fit the data less well than their competitors should be discarded. Equation (5.3) defines the set of structures that will be included in the belief state:

$$\mathcal{W} = \{B_i^s : \frac{max_k\{P(B_k^s|D)\}}{P(B_i^s|D)} \leq c\} \tag{5.3}$$

where c is some constant and k is an index over all the possible BN Structures. As highlighted in (Madigan & Raftery, 1994) the value of c depends on the context. The scores of the BN structures in $\mathcal{W}$ are then used to compute a probability distribution ($P_t(B^s)$) over all structures in $\mathcal{W}$. The set $\mathcal{W}$ does not give a complete epistemic state because a lot of BN structures whose combined score is quite significant would have been discarded. However, the assumption in this approach is, every edge α_{ji} that is entrenched well enough in the epistemic state should be in at least one of the high scoring networks. Thus, by having $\mathcal{W}$ large enough the edge weights can be estimated from the BN structures in $\mathcal{W}$ using Equation (5.4);

$$P(x_j \to x_i|D) = \sum_{B_k^s : B_k^s \models (x_j \to x_i)} P(B_k^s|D) \tag{5.4}$$

where $x_j \to x_i$ is a new notation for the edge/proposition α_{ji} introduced for convenient representation of an edge from X_j to X_i.

Assuming that for a distinct pairs of variables X_j and X_i, the event of X_j having some set of variables as its parents is independent of the event X_i having some set of parents, the probability of a BN structure is given by the product of the probabilities of the parent structures encoded in the BN structure, as shown in Equation (5.5) (Buntine, 1991; Cooper & Herskovits, 1992).

$$P(B^s) = \prod_{1 \leqslant i \leqslant n} P(\pi_i^s \rightarrow x_i) \tag{5.5}$$

Equation (5.5) makes it possible to compute a complete belief state over all the possible consistent BN structures including those which were discarded from $\mathcal{W}$ by Equation (5.3). This implies that Equation (5.2) can be rewritten as Equation (5.6)

$$P(D, B^s) = \prod_{i=1}^{n} P(\pi_i^s \rightarrow x_i) \prod_{j=1}^{q_i} \frac{\Gamma(N'_{ij})}{\Gamma(N'_{ij} + N_{ij})} \prod_{k=1}^{r_i} \frac{\Gamma(Nijk' + N_{ijk})}{\Gamma(N'_{ijk})} \tag{5.6}$$

Assuming that the event x_m being a parent of x_i is independent of the event x_n being a parent of x_i, the probabilities of parent configurations can be computed for all the variables using Equation (5.7) (Madigan & Raftery, 1994)

$$P(\pi_i^s \rightarrow x_i) = \prod_{x_m \in \pi_i^s} P(x_m \rightarrow x_i) \prod_{x_m \notin \pi_i^s} (1 - P(x_m \rightarrow x_i)) \tag{5.7}$$

The $P(x_m \rightarrow x_i)$ probabilities to be used in Equation (5.7) should be derived from the updated belief state using Equation (5.8).

$$P(x_m \rightarrow x_i) = \sum_{B_k^s : B_k^s \vDash (x_m \rightarrow x_i)} P(B_k^s) \tag{5.8}$$

The event conditional probability distributions are computed from the edge weights using the Algorithm 5.1 as follows;

For each of the structures in $\mathcal{W}$, list one-step transitions, which can be either an edge addition, an edge removal, an edge reversal or null event, from one BN structure to some other structure in $\mathcal{W}$. The null event is given a weight of one (1), which is the highest weight possible for any event. The event weight for addition of an edge is taken to be the weight of the edge to be added denoted by $edgeWeight(e.edge)$. The weight of an edge removal event is given by, $1 - (edgeWeight(e.edge)$. For an edge reversal event, we assume an edge reversal to be a combination of two independent events, an edge removal and an edge insertion. The weight for an edge reversal was taken as a product of the weights of an edge removal and an edge addition, $(1 - (edgeWeight(e.edge))(edgeWeight(\overleftarrow{e.edge}))$. The edge $\overleftarrow{e.edge}$ is the reverse of the edge $e.edge$. For each structure, the event conditional probability distribution,$P_t(e|w)$, was calculated as normalised weights of all the possible events. Once the event conditional probability distributions, and the probability distribution over the significantly entrenched Bayesian Network structures are obtained, Belief Update as defined by the model in Equation (4.6) is effected.

Algorithm 5.1 Algorithm for computing the Conditional Event Probability Distributions

Input:$\mathcal{W}$
Output:Conditional Event Probability Distributions for each B_i^s in $\mathcal{W}$
procedure ComputeEventProbDistrs
  for each $B_i^s \in \mathcal{W}$ **do**
    $list \leftarrow$ list all events that maps B_i^s to another BN structure in $\mathcal{W}$, their event type and edge weights
    for $e \in E^s$ **do**
      if $e.eventType == "addition"$ **then**
        $e.eventWeight = edgeWeight(e.edge)$
      else if $e.eventType == "removal"$ **then**
        $e.eventWeight = 1 - edgeWeight(e.edge)$
      else if $e.eventType == "reversal"$ **then**
        $e.eventWeight = (1 - edgeWeight(e.edge)) * (edgeWeight(\overleftarrow{e.edge}))$
      else if $e.eventType == "null"$ **then**
        $e.eventWeight = 1$
    $sumEventWeights = \sum_{j=1}^{m} e_j.eventWeight : B_i^s \xrightarrow{e_j} B_k^s, \forall B_k^s \in \mathcal{W}$
    for $e \in E_s$ **do**
      $e.conditionalProbability = e.eventWeight/sumEventWeights$

5.3 A Toy Example to illustrate the Model

This section presents a demostration of how the Belief Change Operator defined in Section 5.2 works. The toy example introduced in Chapter 4 will be used. The data used in this section was generated by the researcher with the sole aim of illustrating how the *UBCOBaN* operator works. Hence this Chapter does not seek to validate any claims about the model and the operator, except to show how the operator works.

Suppose a probability distribution over the possible BN structures in $\mathcal{W}$ (that fit the data well enough) at time t is as shown in Figure 5.1. Given this distribution, the edges' weights can be computed using Equation (5.8). The edge weights are shown in Table 5.1.

Table 5.1.: Edge weights computed from Figure 5.1

Edge	**Weight calculation**	**Weight**
$RA(p) \rightarrow Acpted(p,j)$	0.5+0.25+0.15+0.03+0.02	0.95
$RA(p) \rightarrow GP(p)$	0.5+0.05+0.03+0.02	0.35
$GP(p) \rightarrow Acpted(p,j)$	0.5+0.25+0.15+0.05	0.95
$DBR(j) \rightarrow Acpted(p,j)$	0.5+0.25+0.15+0.05+0.03+0.02	1
$GP(p) \rightarrow RA(p)$	0.15	0.15
$Acpted(p,j) \rightarrow GP(p)$	0.02	0.02

The edge weights in Table (5.1) are a reflection of how much the edges (i.e. a sentence) are entrenched in the epistemic state. An edge with a weight of one (1) (e.g. $DBR(j) \rightarrow Acpted(p,j)$) is maximally entrenched and hence it is in the Belief Base. Without loss of generality, if a Belief Base, K, is defined as the top of the p-function as follows; $K = A : P_t(A) \geq 0.95$, the Belief Base resulting from Table 1 is; $K = \{RA(p) \rightarrow Acpted(p,j), GP(p) \rightarrow Acpted(p,j),\ DBR(j) \rightarrow Acpted(p,j)\}$. This represents the theory that acceptance of a paper in a journal depends on the reputation of the author(s) of the paper, quality of the paper, and whether the journal uses double-blind review.

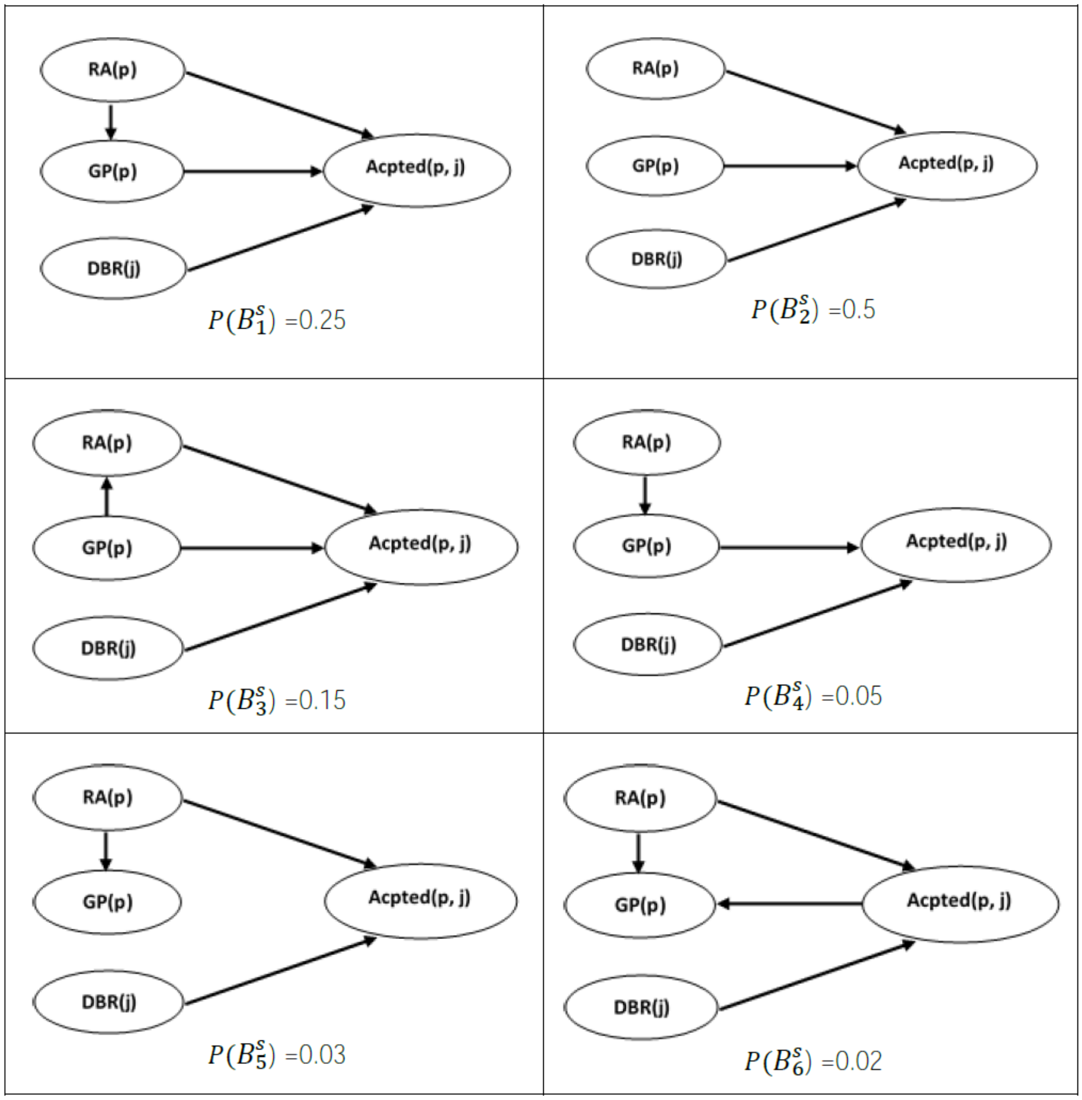


Figure 5.1.: BN structures in the set of structures that fit the data well enough and their probability distributions.

5.3.1 Updating the Epistemic State

As discussed in Section 5.2, the operator starts with the update function and then do revision on the updated epistemic state. In order to do the update operation, the operator needs model parameters for update. These are computed using the algorithm in Algorithm 1. Table (5.2) shows the *event probability distributions* for each BN structure.

Using Equation (4.6) and skipping all computations with $P(B^{s\prime}|B^s, e) = 0$ and their corresponding $P(e|B^s)$ the new updated distribution of the BN structures in $\mathcal{W}$ at time $t+1$ is shown in Table (5.3). The table shows the update function redistribute the probability mass to favour BN structures that are more likely to result from non-null event transitions with higher weights. In fact, it takes the probability mass from BN structures which are more likely to experience transitions and distributes the probability mass to the BN structures that results from these transitions. In Table (5.3), as a results of the fact that transitions to B_1^s from all the other BN structures have higher probabilities, the probability mass is shifted from the other BN structures to B_1^s.

Table 5.2.: Event probability distributions for each Structure in $\mathcal{W}$

BN Structure	Event	Resultant State	Event Weight	Event Probabilities
B_1^s	Null event	B_1^s	1	10000/18485
	$Rem(RA(p) \rightarrow GP(p))$	B_2^s	1-0.35 = 0.65	6500/18485
	$Rev(RA(p) \rightarrow GP(p))$	B_3^s	(1-0.3)(0.15)=0.0975	975/18485
	$Rem(RA(p) \rightarrow Acpted(p,j))$	B_4^s	1-.0.95 = 0.05	50/18485
	$Rem(GP(p) \rightarrow Acpted(p,j))$	B_5^s	1-0.95 = 0.05	50/18485
	$Rev(GP(p) \rightarrow Acpted(p,j))$	B_6^s	(1-0.95)(0.02) = 0.001	1/18485
B_2^s	$Add(RA(p) \rightarrow GP(p))$	B_s^s	0.35	35/150
	Null event	B_2^s	1	100/150
	$Add(GP(p) \rightarrow RA(p))$	B_3^s	0.15	15/150
B_3^s	$Rev(GP(p) \rightarrow RA(p))$	B_1^s	(1-0.15)(0.35)=0.2975	2975/21475
	$Rem(GP(p) \rightarrow RA(p))$	B_2^s	1-0.15 = 0.85	8500/21475
	Null event	B_3^s	1	10000/21475
B_4^s	$Add(RA(p) \rightarrow Acpted(p,j))$	B_1^s	0.95	95/195
	Null event	B_4^s	1	100/195
B_5^s	$Add(GP(p) \rightarrow Acpted(p,j))$	B_1^1	0.95	95/197
	Null event	B_5^s	1	100/195
	$Add(Acped(p,j) \rightarrow GP(p))$	B_6^s	0.02	2/195
B_6^s	$Rev(Acped(p,j) \rightarrow GP(p))$	B_1^s	(1-0.02)(0.95) = 0.931	931/2911
	$Rem(Acped(p,j) \rightarrow GP(p))$	B_5^s	1-0.02 = 0.98	980/2911
	Null event	B_6^s	1	1000/2911

The sum of the probabilities in Table (5.3) sum up to one (1) confirming that the update function results in a consistent p-function. The updated edge weights in Table (5.4) were computed from the updated Belief State in Table (5.3) using Equation (5.7). Table (5.4) shows that after update the weights of the edges appearing in higher scoring BN structures get more entrenched and those appearing in low scoring BN structures get less entrenched.

Table 5.3.: Updated belief state

BN Structure	$P_t(B_i^s)$	Calculation	$P_{t+1}(B_i^s)$
B_1^s	0.25	(0.25)(10000/18485)+0.5(35/150) $+\cdots+$ 0.02(931/2911)	0.31791
B_2^s	0.5	(0.25)(6500/18485)+(0.5)(100/150) + 0.15(2500/21475)	0.48061
B_3^s	0.15	(0.25)(975/18485)+(0.5)(15/150)+(0.15)(10000/21475)	0.13304
B_3^s	0.05	(0.25)(50/18485)+(0.05)(100/195)	0.03241
B_5^s	0.03	(0.25)(50/18485)+(0.03)(100/197)+(0.02)(2/197)	0.02872
B_6^s	0.02	0.25(1/18485)+(0.03)(2/197)+(0.02)(1000/2911)	0.00731

Table 5.4.: Updated Edge Weights

Edge	Weight calculation	Weight
$RA(p) \rightarrow Acpted(p, j)$	0.31791+0.48061+0.13304+0.02872+0.00731	0.9676
$RA(p) \rightarrow GP(p)$	0.31791+0.03241+0.02872+0.00731	0.3864
$GP(p) \rightarrow Acpted(p, j)$	0.31791+0.48061+ 0.13304+0.03241	0.964
$DBR(j) \rightarrow Acpted(p, j)$	0.31791+0.48061+0.13304+0.03241+0.02872+0.00731	1
$GP(p) \rightarrow RA(p)$	0.13304	0.133
$Acpted(p, j) \rightarrow GP(p)$	0.00731	0.0073

Using the edges in Table (5.4) and their weights, we can generate all the parent structure configurations with non-zero probabilities. The resulting parent structure configurations and their distributions for each node are shown in Table (5.5). The fact that the sum of the probabilities of the parent structures for each node sum to one (1) serves to confirm the correctness of Equation (5.6). For the variable $Acpted(P, J)$, the parent structure configurations, $\{\}$, $\{RA\}$, $\{GP\}$, $\{RA, GP\}$ were omitted since they give a probability of zero. This was a result of the fact that the weight of $DBR(j) \rightarrow Acpted(p, j)$ is one (1) and any parent configuration which does not have $DBR(j)$ will have a probability of zero. This is very logical from a Belief Change perspective. Any worlds (BN structures) that contradict the proposition $DBR(j) \rightarrow Acpted(p, j)$ should not be accepted after belief update since the proposition is maximally believed. Contradictions to this proposition can be accepted only after revision if data inconsistent with the proposition is observed.

Table 5.5.: Parent configurations for each node in the domain and their probabilities. *for Acpted(P,J) parent configurations, {}, {RA}, {GP}, {RA, GP} were omitted since they give a probability of zero

	Variables							
	RA		GP		DBR		Acpted(P,J)*	
	π_i	$P(\pi_i)$	π_i	$P(\pi_i)$	π_i	$P(\pi_i)$	π_i	$P(\pi_i)$
Parent Structure	None	0.86696	None	0.60916	None	1	DBR	0.00117
	GP	0.13304	Acc	0.00449			RA,DBR	0.03487
			RA	0.38353			GP, DBR	0.03124
			RA, Acc	0.00282			RA,GP,DBR	0.93273
Total		1		1		1		1

5.3.2 Revising the Epistemic State

From Table 5.5, (2)(4)(1)(4) = 32 BN structures with non-zero probabilities can be generated, though not all of them will be consistent BN Structures. The explosion of structures may seem a disadvantage from a computational perspective, but it is key to belief change on the dependences in the BN structure, because it allows new BN structures to be added into the belief state and evaluated on how well they fit the observations. Intuitively all the consistent BN structures among 32 BN structures can be generated and then used in the revision phase of the operator. However, the explosion of the BN structures is often prohibitive hence in practical implementation of the belief change operator, heuristics are advocated. The Search-and-Score algorithm is applied for revision and the updated belief state, will give the structure priors to be used by the Search-and-Score algorithm.

For the revision part of the Belief Change operator the synthetic data in Table (5.6) was used. All variables were considered to be binary, and the data was hand-crafted by the authors only for the purposes of illustrating how the proposed Belief Change operator works. The revision function used in this illustration is based on the K2 algorithm (Cooper & Herskovits, 1992) given in Equation (5.9). In reality, for large Networks any of the Bayesian-based Search-and-Score structure learning algorithms can be used, provided that they use the updated structure priors in revising the belief state.

$$P(D, B^s) = \prod_{i=1}^{n} P(\pi_i^s \rightarrow x_i) \prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(N_{ij} - r_{ij} - 1)!} \prod_{k=1}^{r_i} N_{ijk} \tag{5.9}$$

where, n, q_i, r_i, N_{ijk}, and N_{ij} as defined in Theorem 4.1.

Table 5.6.: The dataset used to illustrate revision

Case	Variables			
	$RA(P)$	$GP(P)$	$DBR(J)$	$Acpted(P,J)$
1	yes	Yes	yes	Yes
2	no	No	no	No
3	no	No	yes	No
4	yes	No	no	No
5	yes	Yes	yes	Yes
6	yes	Yes	no	No
7	no	No	no	No
8	no	Yes	yes	Yes
9	yes	Yes	yes	Yes
10	no	No	no	No
11	no	No	yes	No
12	no	No	no	Yes

The revised Belief State obtained from applying Revision without Update and Revision after Update are shown in Table (5.7). The new highest scoring BN structure is now B_1^sin both cases. This is a result of the evidence in the data supporting the proposition, $RA(p) \rightarrow GP(p)$. The proposition $RA(p) \rightarrow Acpted(p,j)$ is being gracefully dropped to make way for the proposition, $RA(p) \rightarrow GP(p)$. Assuming the top of the p-function is defined as $P(\alpha_{ji} \geq 0.95)$, Table (5.8) shows that Belief Base (K) from the operator defined in this chapter still has three(3) propositions, whereas the one from classical structure learning without update now has two propositions, $K = \{GP(p) \rightarrow Acpted(p,j), DBR(j) \rightarrow Acpted(p,j)\}$. A closer look at the edge weights in Table (5.8) shows that the *UBCOBaN* seems to be more stubborn on giving up beliefs, but is faster to accept new beliefs than the structure learning operator without Update. This, however, is still to be validated through experimentation. The focus in this chapter was to illustrate how the proposed belief change operator works. It is important to note that even through the proposed belief change operator did not change the Belief Base, the underlying belief state has been significantly changed. The belief in B_1^s being the current true network structure has been increased from 0.25 to 0.625 and that of B_2^s has been reduced from 0.5 to 0.162. The edge weights have also been changed, with the weight of the edges being supported by the data bearing more significant change. The weight of the edge $RA(p) \rightarrow GP(p)$ increased from 0.35 to 0.6856 under the proposed operator.

Table 5.7.: Revised Belief State

BN Structure	$P_t(B_i^s)$	$P_{t+1}(B_i^s)$ without update	$P_{t+1}(B_i^s)$ with update
B_1^s	0.25	0.56195	0.62515
B_1^s	0.5	0.17029	0.16202
B_1^s	0.15	0.18417	0.15237
B_1^s	0.05	0.0631	0.04466
B_1^s	0.03	0.01972	0.01558
B_1^s	0.02	0.00078	0.00022

Table 5.8.: The edge weights after revision

		Revised Edge Weights	
Edge with update	Original Edge Weights	without Update	with Update
$RA(p) \rightarrow Acpted(p, j)$	0.95	0.9369	0.9553
$RA(p) \rightarrow GP(p)$	0.35	0.6455	0.6856
$GP(p) \rightarrow Acpted(p, j)$	0.95	0.9795	0.9842
$DBR(j) \rightarrow Acpted(p, j)$	1	1	1
$GP(p) \rightarrow RA(p)$	0.15	0.1842	0.1524
$Acpted(p, j) \rightarrow GP(p)$	0.02	0.0008	0.0002

5.4 Conclusions

This chapter presented the Unified Belief Change Operator for Bayesian Networks. The operator is one instance of the Unified Belief Change Model for Bayesian Networks defined in Chapter 4. The chapter discussed how the Conditional Event Probability distributions to be used by the operator are computed using the edge weights and the propensity of a transition from one BN structure to the other in the set of possible BN structures.

To illustrate how the operator works, a toy example, modelling beliefs about paper acceptance for publication in journals, was used. The illustration showed that the modelling of event probabilities used in this study adheres to the principles of probability calculus. This was shown by the fact that parent configuration for each of the variables in the BN structures after Update with the computed event probabilities summed up to 1 (see Tables (5.3) and (5.5)). The illustration also showed that the *UBCOBaN* operator is able to redistribute probability mass within the Belief State from the BN structures less consistent with the data to the BN structures more consistent with data.

6. EMPERICAL EVALUATION OF THE *UBCOBAN* USING BENCHMARK BAYESIAN NETWORKS

"Data is a precious thing, and
will last longer than the systems themselves.
Tim Berners-Lee

6.1 Introduction

Having defined the Unified Belief Change Model for Bayesian Network Structures and its corresponding operator, this chapter presents an empirical evaluation of the model and the operator in propositional Bayesian Networks. The evaluation of the model in First Order Bayesian Network structures is deferred to Chapter 7. The goal of the evaluation was to substantiate the claim being made in this thesis that use of belief change principles in evolving BN structures results in rational belief change of Bayesian Network Structures. To this end, the evaluation endeavoured to establish that: (i) *UBCOBaN* does not unnecessarily change the structure of a Bayesian Network (i.e. it fulfills the principle of minimal change); (ii) *UBCOBaN* has the agility to shift the network structure towards the true network structure, whenever there are observations inconsistent with the current structure. Benchmark Bayesian Network structures were used to evaluate the operator. The main metric used to evaluate the operator is the Structured Hamming Distance (SHD) (Tsamardinos, Brown, & Aliferis, 2006). Alternatively, structure score metrics such as the DBeu, KL divergence, MDL could be used, but they are known to usually favour the algorithms that used them in learning the network structure

In all the experiments conducted in this and the next chapter, the performance of *UBCOBaN* was benchmarked against the performance of the classical Search-and-

Score algorithm implemented in the *Banjo* API[1] (which will hence forth be referred to as the the *Banjo* operator).

6.2 Benchmark Bayesian Networks

The benchmark Bayesian Networks that were used for evaluation in this chapter are as follows:

1. the **ASIA Network** (Lauritzen & Spiegelhalter, 1990),
2. the **ALARM Network** (Beinlich et al., 1989)
3. the **HAILFINDER Network** (Abramson et al., 1996)
4. the **HEPAR II Network** (Onisko, 2003), and
5. the **ANDES Network** (Conati et al., 1997)

The ASIA (Lauritzen & Spiegelhalter, 1990) is a synthetic Bayesian network with 8 nodes and 8 edges. It is the smallest benchmark network used in this study. The ALARM Network (Beinlich et al., 1989) is the most widely used benchmark Bayesian Network for learning network structure in Propositional Bayesian Networks. It was originally described by Beinlich et al (1989 as a network for monitoring patients in intensive care. The ALARM Network as defined in (Beinlich et al., 1989) consists of 37 nodes of two, three or four states, and 46 edges. The HAILFINDER Network (Abramson et al., 1996) is a Bayesian network designed to forecast severe summer hail in north eastern Colorado. It has 56 variables and 66 edges. The HEPAR II Network (Onisko, 2003)was designed as a probabilistic graphical causal model for diagnosis of liver disorders for use in both clinical practice and medical training. The HEPAR II network has 70 variables and 123 edges. The ANDES Network was designed as a model for an Intelligent Tutoring System for Newtonian physics. The network has 223 nodes and 338 edges.

[1]https://users.cs.duke.edu/ amink/software/banjo/

6.3 Implementation of *UBCOBaN*

UBCOBaN was implemented in Java using the *Banjo* API [2] as the underlying API to provide the classical Search-and-Score BN Structure Learning algorithms. The core architecture of *Banjo* is shown in Figure 6.1.

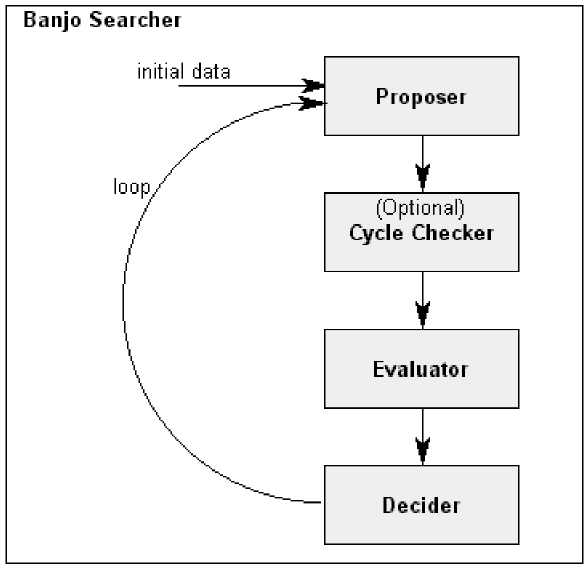


Figure 6.1.: The core *Banjo* Objects.

This section discusses the core *Banjo* components that were modified for the algorithm to work as a Belief Change operator for structure in Bayesian Networks. *Banjo* implements the following generic components for Search-and-Score algorithms:

1. **Searcher** : The Searcher is the top level component of the *Banjo* architecture that implements how a BN structure learning algorithm searches through the space of possible structures to find a BN structure that best fits the data. For any BN structure learning algorithm, a searcher is built around a search loop that executes for an allotted amount of time or until a specified number of networks have been proposed and considered. In order for the searcher to explore the BN structures in the search space, it uses a **Proposer** component to

[2]https://users.cs.duke.edu/ amink/software/banjo/

suggest alternative BN structures. To make a decision on whether to accept the proposed BN structure the **Evaluator** and the **Decider** components are used. A proposed network is only accepted by the Decider, when its score computed by the **Evaluator** is better than the one of the current best network.

2. **Proposer**: The proposer is responsible for selecting a change (or a list of changes) based on whether to add, delete, or reverse an edge in the current network. In *Banjo* the proposer implements two similar functions: one provides a single change (named suggestBayesNetChange) and one provides a list of changes (named suggestBayesNetChanges). The function of suggestBayesNetChange is to implement the RandomLocalMove proposer and the suggestBayesNetChanges is used to implement the AllLocalMoves proposer. The differences between the RandomLocalMove and AllLocalMoves proposer is that in the former the searcher evaluates the efficacy of resulting BN structure after a single local change, while in the latter the search evaluates the BN structure after a group of all the possible local moves.

3. **Evaluator**: The Evaluator computes the score of a network, based on some scoring metric. *Banjo* currently implements only one Evaluator, which uses the BDe metric to compute a network's score, as described by Cooper and Herskovits (1992). This is one of the components which was modified in this study to enable the scoring metric to use the updated Belief State parent structure priors.

4. **Decider**: A Decider in *Banjo* determines whether the proposed network in the current search iteration will be accepted as the new current network for the next iteration, or if it will be rejected, in which case the search proceeds from the current network.

Implementation of *UBCOBaN* was done through adapting how some of the components above work and introduction of additional components to cater for the Belief update component and management of Epistemic States. The decider and the proposer components remained the way they were implemented in *Banjo*. The constructor for Searcher component was only slightly modified for it to accept edgeweights from the Epistemic State as parameters for the search process. The edgeweights were to be used by the evaluator component. The evaluator component was modified as discussed in Section 5.2.2 such that it uses Equation 6.1 to compute the Bayesian Dirichlet (BDe) metric for the proposed BN structures.

$$P(D, B^s) = \prod_{i=1}^{n} P(\pi_i^s \rightarrow x_i) \prod_{j=1}^{q_i} \frac{\Gamma(N'_{ij})}{\Gamma(N'_{ij} + N_{ij})} \prod_{k=1}^{r_i} \frac{\Gamma(Nijk' + N_{ijk})}{\Gamma(N'_{ijk})} \tag{6.1}$$

The BDe metric as implemented in *Banjo* assumes uniform structure priors. By so doing the searcher disregards the use of prior information on the BN structures in changing the current BN structure in response to the data observed. The modification done to the evaluator for it to implement Equation 6.1 was therefore necessary to enable the searcher to use prior structure information in the form of edge weights to compute the DBe scores.

6.3.1 Implementation of the Data Models for the Epistemic Space

An XML data model was defined to hold the Epistemic State at any given time. XML was a natural choice for the data model to enable the Epistemic State to be sharable over the network across different administrative domains. The XML schema for the data model is shown in Appendix A. Appendix B shows an except of an Epistemic State for the ASIA network. A utility component for reading data from and writing data to the Epistemic State file, EpistemicStateProcessor was implemented and added to the *Banjo* utility components. This utility component uses JAXB for marshalling Java objects into XML and and vice-versa. Another utility component which was added for handling of uncertainty in the model is the EdgesLikelihoodComputer component. This component was responsible for computing the edgeWeights for a given belief state. It was implemented as a utility component to enable it to be used after both Belief Update and Belief Revision operations.

6.3.2 Implementation of the Belief Update Model

The updater component responsible for Belief Update was implemented taking advantage of what is existing in the *Banjo* API and the afore-mentioned components. The component computes the updated Epistemic State using the *Conditional Event Probability distribution* and the Belief State, a distribution over all the possible networks. Thus, there was need for computing *Conditional Event Probability distribution*. Thus, apart from implementing the update process, this component also implemented Algorithm 5.1 discussed in Chapter 5. Figure 6.2 shows the overall architecture of *UBCOBaN*.

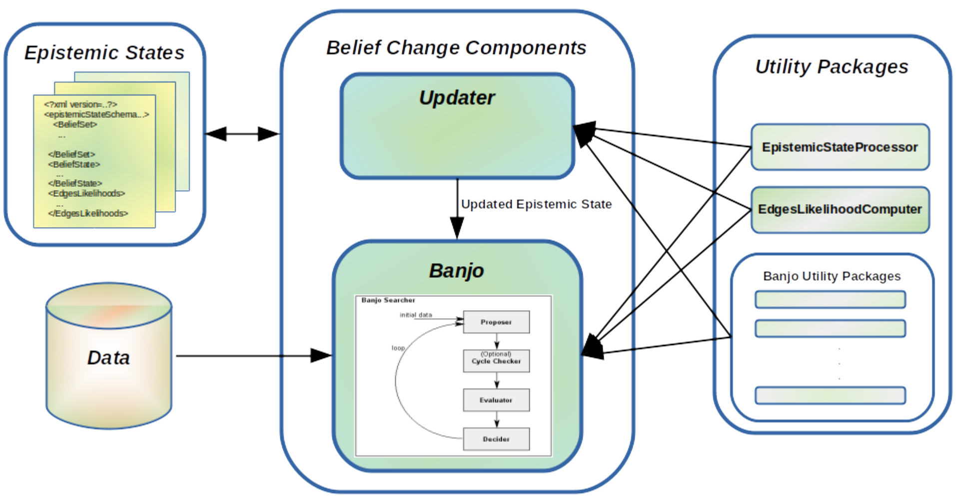


Figure 6.2.: *UBCOBaN* Core Components.

6.4 Evaluation of the Unified Belief Change Model

For all the experiments carried out in this study, a greedy searcher was used to search for the highest scoring BN structure from the space of all possible structures. The following metrics were used to evaluate the operator (de Jongh & Druzdzel, 2009)

Missing Edges (ME): counts the number edges that are in the target BN structure, but are not in the learned BN structure.

Extra Edges (EE): counts the number of edges that are present in the learned structure but are not in the target BN structure.

Correct Edges(CE): counts the number of edges that are both in the target BN structure and the learned structure, regardless of their orientation.

Correct Edge Direction (CED): counts the number of directed edges in the learned structure that have the same orientation with some of the edges in the target BN structure.

Incorrect Edge Directions (ICED): counts the number of directed edges in the learned structure that are oriented incorrectly when compared with the target BN structure.

Structural Hamming Distance (SHD): Counts the number of changes that have to be made to the learned BN network for it to turn into the target BN structure. It is the sum of measures; Missing Edges, Extra Edges, and Incorrect Edge Directions.

$$SHD = ME + EE + ICED \tag{6.2}$$

Recall: gives the ratio of correctly identified edges to the total number of edges which were suppose to be identified. It is a measure of how good the structure learning algorithm is in identifying all the correct edges.

$$Recall = \frac{CED}{ME + CED + ICED} \tag{6.3}$$

Precision: it gives a ratio of the correctly identified edges to the total number of edges identified. it is a measure of how good the structure learning algorithm is at returning only the correct edges.

$$Precision = \frac{CED}{EE + CED + ICED} \tag{6.4}$$

For computation of precision and recall, the Correct Edge Directions measure was used instead of the Correct Edges. This was meant to make sure that Recall and Precision are both based on the intuition used in the definition of the SHD metric. Even though in terms of conditional probability calculus, an Incorrect Edge Direction does not result in a different probability distribution from the target BN as long as the v-structures are maintained, the SHD still considers the cost of ICED in its computation.

For all experiments that needed data simulation, the bnlearn package [3] was used for data simulation. The Rserve[4] package was used to enable the java program to communicate with R for data simulation. All data simulation was done from the benchmark BNs using the bn.fit models stored in bnlearn.

6.4.1 Evaluating the Minimal Change Property of *UBCOBaN*

The minimal change property of *UBCOBaN* was evaluated through analysis of its *accuracy* and *stability* benchmarked against the the classical Search-and Score algorithm implemented in *Banjo*. Two sets of experiments were carried out.

The first set of experiments investigated the accuracy and stability of *UBCOBaN* in terms of the returned Bayesian Networks deviating from the BN structure emitting the data being used to effect Belief Change. This was achieved through repeatedly sampling data from the gold BN structure and using the simulated data to effect Belief Change on BN structure from the previous run. Accuracy and Stability of the the resulting BN structures were then evaluated. Accuracy was measured in terms of

[3]http://www.bnlearn.com/

[4]https://cran.r-project.org/web/packages/Rserve/index.html

the SHD of the resulting BN structure from the BN structure simulating the data. Stability of the Belief Change operators was evaluated in terms of how much the resulting BN structure deviated from the BN structure from the previous run. Stability deals with ascertaining how sensitive the predictions of a machine learning models are to small changes in the training data. A stable learning algorithm would produce almost similar results for small changes in the training data. Owing to the fact that stable learning algorithms return almost consistent results, stability plays a crucial role in entrenching users' trust in the algorithms. In our case, stability is characterised in terms the dispersion of the SHDs of the retained current BN structure from the one that resulted from the previous run. Thus, our conceptualisation of stability is characterised as the distribution of pairwise similarities between the BN structure obtained from the current run compared to the BN structure from the previous run. This conceptualisation is similar to how stability is evaluated for clustering algorithms in (Ben-Hur, Elisseeff, & Guyon, 2002; Lange et al., 2004). However, Stability of Belief Bases was evaluated in terms of the the dispersion of the SHDs of the Belief Bases from the gold BN structure simulating the data. In this case, a stable Belief Change Operator returns almost the same Belief Base for each run. This implies that the dispersion of the SHDs of the resulting BN structures from their corresponding gold BN structures will be almost zero.

The second set investigated the effect of the sample size on the accuracy and stability of the evolved BN structures. This was achieved by varying the sample sizes and observing the effect of the increase of the sample size on the accuracy and stability of the resulting BN Structures.

For each of the experiments discussed above the afore-mentioned metrics were recorded comparing:

i. the Belief Base (Edges with a probability of 1) to the gold BN structure.

ii. the returned BN structure to the gold BN structure, and

iii. the returned BN structure to the prior BN structure.

6.4.1.1 Experimental Setup

In the first set of experiments, 30 runs of incremental application *UBCOBaN* on the BN structure that resulted from the previous run were conducted. The data used in each run would be simulated from the gold BN structure. For the first run, belief change was performed on the gold standard Bayesian Network using the data simulated from the gold standard Bayesian Network. The goal here is two-fold. First, we would want to evaluate how much the returned BN structure and its Belief Base would have deviated from the gold BN structure after a considerable number of Belief Change iterations. To this end, the average metrics for the last 5 of the 30 runs were used to check how much the best BN structures and their corresponding Belief Bases deviated from the gold BN structure. Second, we would want to investigate whether the Belief Change Operators consistently returned almost the same best BN structure for all 30 runs conducted. This was achieved by analysing both the SHD of the returned BN structure from the BN structure returned by the previous run, and the dispersion of the same SHDs. Small SHD values imply stability of the operator and large values instability. If the operators were perfectly stable they would return the same BN structure for each run with a constant SHD from the BN structure returned from the previous run. Thus, the dispersion of the SHDs would be zero.

In the second set of experiments, the sample sizes were being incremented by 2000 units starting from 200 samples. For each sample size 10 runs were conducted. The samples were simulated from the gold BN structure. The BN structure from the previous run was taken to be the prior network for each run. The ME, EE, CED, ICED, SHD, Recall and Precision metrics of the best BN structure from the gold BN structure, best BN structure from the prior BN Structure, and the Belief Base from the gold BN structure were recorded. Accuracy and stability for both the best BN structures and the Belief Bases were evaluated as discussed above for each sample size.

6.4.1.2 Experimental Results: Accuracy and Stability *UBCABaN* based on Belief Bases

This section discusses the results of an investigation of the accuracy and stability of the Belief Bases (edges with a probability of 1) returned by *UBCOBaN*.

Tables 6.1 and 6.2 show a comparison of *UBCOBaN* and *Banjo* with respect to how close to the gold standard the Belief Bases they returned are using the metrics; Missing Edges, Extra Edges, Correct Edge Direction, Correct Edges, Structural Hamming Distance, Recall and Precision. The metrics shown in the table are an average of the last 5 runs of the 30 runs of incremental application of the belief change operators.

Table 6.1.: Average distance of Belief Bases from the gold BN Structures with respect to ME, EE, CED, and CE

	ME		EE		CED		CE	
	UBCOBaN	*Banjo*	*UBCOBaN*	*Banjo*	*UBCOBaN*	*Banjo*	*UBCOBaN*	*Banjo*
ASIA	1	4.2	0	0	6	3.6	7	3.8
ALARM	1	5	0	0	43	41	45	41
HAILFINDER	20.6	13.4	2.6	2.8	45.4	52.4	45.4	52.6
HEPAR II	11.4	25.4	0.8	2.4	110.6	95.6	111.6	97.6
ANDES	0.4	10.6	0.4	1.6	337.6	326.3	337.6	327.4

Table 6.2.: Average distance of Belief Bases from the gold BN Structures with respect to SHD, Recall, and Precision

	SHD		Recall		Precision	
	UBCOBaN	*Banjo*	*UBCOBaN*	*Banjo*	*UBCOBaN*	*Banjo*
ASIA	2	4.4	0.75	0.45	0.857	0.933
ALARM	3	5.2	0.935	0.887	0.956	0.995
HAILFINDER	23.2	16.4	0.688	0.794	0.946	0.949
Hepar II	13.2	29.8	0.899	0.777	0.984	0.957
ANDES	0.8	12.8	0.999	0.967	0.999	0.993

The results reflect that *UBCOBaN* returned Belief Bases that are closer to the gold BN structures than *Banjo* in all the benchmark BNs used, except for the HAILFINDER network where *Banjo* performed much better than *UBCOBaN* in all metrics. For the HAILFINDER network, *Banjo* managed to return, on average, belief bases with 80.3% (53 out of 66) of the edges in the gold BN, with an average SHD of 16.4 compared to *UBCOBaN* which returned about 69% (45 out of 66) of the edges in the gold BN, with an average SHD of 23.2. This was due to the fact that *UBCOBaN* was returning 300 BN structures, the set maximum number of best networks, whereas *Banjo* was always returning less than 100 BN structures even though the maximum number of best networks was set to 300. The net effect of the 300 best networks returned by *UBCOBaN* was that some probability mass which would have otherwise be assumed by some edges if less than 100 BN structures were returned was now being distributed to other less entrenched edges that appear in the other 200 less entrenched BN structures ranked in positions 101 to 300. Re-running *UBCOBaN* with the *nBestNeworks*[5] set to 100 resulted in *UBCOBaN* returning Belief Bases closer to the target network than *Banjo*. The researcher hoped that the discarding of BN structures that do not fit the data well using Equation (5.3) discussed in Section 5.2 would address this problem. This suggests that the hyperparameter for the maximum number of BN Structure to be returned, and the c parameter need to be carefully considered for each Benchmark Bayesian Network.

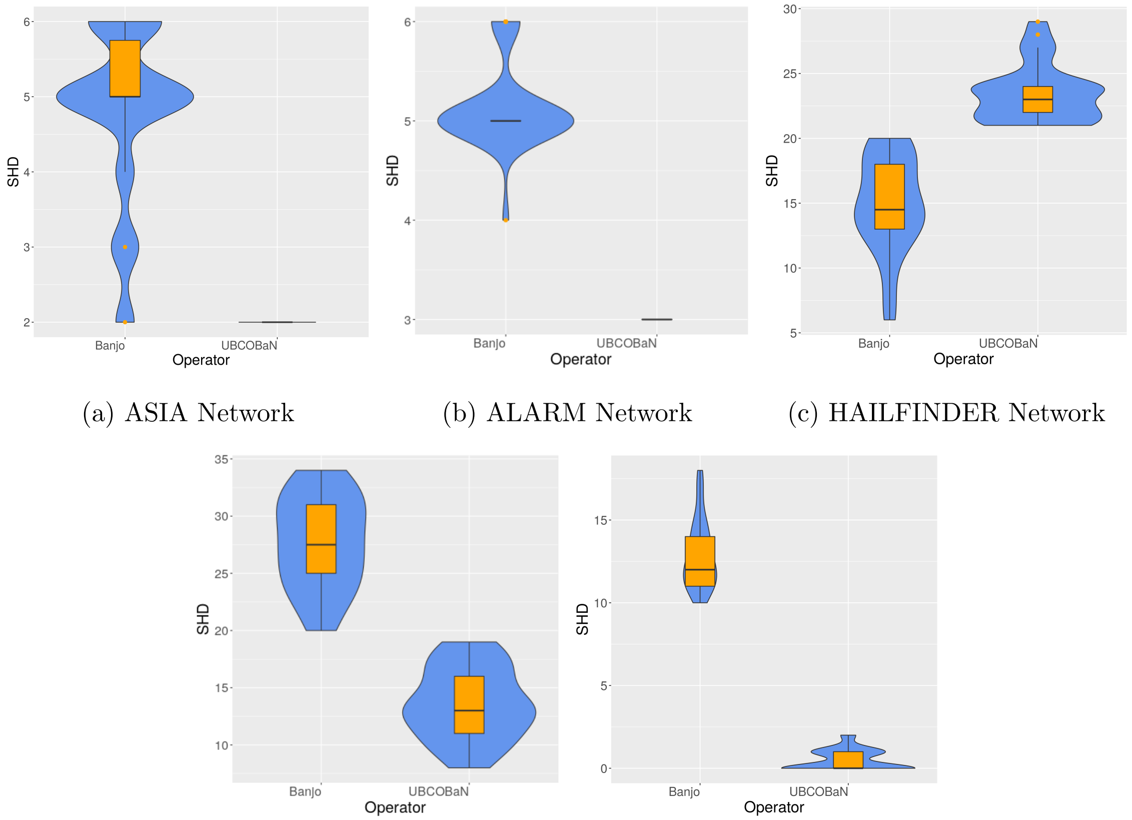
To ascertain the stability of *UBCOBaN* compared to *Banjo* based on the SHD of the Belief Bases returned from the gold BN structure, all 30 runs were used to plot superimposed violin and boxplots plots. These give a view of how the SHD values are distributed. Here stability was analysed relative to the sensitivity of the SHD of the Belief Bases returned by the operators from the gold BN structure simulating the samples. The results of the analysis are shown in Figure (6.3). As was shown in Table 6.2 the Belief Bases returned by UBCOBaN were generally closer to their

[5]nBestNetworks is a hyperparameter used in the *Banjo* API to set the maximum number of structures that a structure learning algorithm should return

corresponding gold BN structure than those returned by *Banjo* for all the benchmark networks except for the HAILFINDER network. However, the violin plots show that the SHDs of *UBCOBaN* from the the gold BN structures are more densely distributed over a small range of values. This indicates that *UBCOBaN* was less sensitive to the changes in the data used to effect Belief Change provided the data was simulated from the gold BN structure the Belief Change Operator seeks to rediscover. Although the SHDs of the Belief Bases returned by *UBCOBaN* from the gold BN structure for the smaller networks (ASIA and ALARM networks) were not necessarily zero, their dispersion was almost zero. This indicates that *UBCOBaN* is stable for all the networks considered in the experiment. *UBCOBaN* was found to be less stable on the HEPAR II and the HAILFINDER network, but it still had better stability when compared to *Banjo*. The Belief Bases returned by *UBCOBaN* on the ANDES network were relatively closer to the gold BN structure than on all the other networks. However, the dispersion was a bit higher than that for the ASIA and ALARM networks. Given the size of the ANDES network, one would have expected Belief bases from ANDES network to have the worst SHD from the gold BN structure than all the other networks. This surprising result is most likely due to some factor inherent in the benchmark ANDES network used to simulate the data. This hypothesis seems to be supported by the fact that *Banjo* also gets relatively good results for a Network of its size. This could be due to the quality of the parameters defined for the network.

6.4.1.3 Experimental Results: Accuracy and Stability of UBCOBaN based on Best BN Structures

Tables (6.3) and (6.4) show a comparison of the SHDs of the best BN structures from the gold BN networks. The results shown in the tables reflect the average of the last 5 runs of the 30 runs conducted for each benchmark BN. *UBCOBaN* produced BN structures which are closer to the gold BN than BANJO for all the benchmark



(a) ASIA Network (b) ALARM Network (c) HAILFINDER Network

(d) HEPAR II Network (e) ANDES Network

Figure 6.3.: Stability of UBCOBaN based on the dispersion of SHD of the Belief Bases from the gold BN structure

networks except for the ASIA network, where the SHD from gold BN structure was 2 for both *UBCOBaN* and *Banjo*. On the ASIA network, *Banjo* was also found to be better in returning most of the edges in the gold BN structure (higher recall) and not returning edges that are not in the gold BN structure (higher precision) than *UBCOBaN*. A closer look at the other metrics for the ASIA network revealed that on average *UBCOBaN* returned slightly more Correct Edges than *Banjo*. However, its SHD, recall and precision were adversely affected by Incorrect Edge Directions. Thus, in terms of representation of the statistical regularities in the simulated data, *UBCOBaN* returned BN structures closer to the gold BN structures than *Banjo*. This is because, as long as an edge reversal does not result in the introduction or removal of v-structures, the BN with reversed edges represents the same distribution as the original BN. The BN with reversed edges and the original BN are said to be *Markov Equivalent* (Niinimaki & Parviainen, 2012; Verma & Pearl, 1991).

Table 6.3.: Average distance of Best BN Structures from the gold BN Structures with respect to ME, EE, CED, and CE

	ME		EE		CED		CE	
	UBCOBaN	*Banjo*	*UBCOBaN*	*Banjo*	*UBCOBaN*	*Banjo*	*UBCOBaN*	*Banjo*
ASIA	0.6	0.8	0	0.6	6	6.6	7.4	7.2
ALARM	1	0.8	0	1	43	42.6	45	45.2
HAILFINDER	0	3.2	3	9.6	64	61.2	66	63
HEPAR II	1.4	8.4	3.6	10.2	120.6	107.8	121.6	114.6
ANDES	0	2	1	8	338	333.2	338	336

Table 6.4.: Average distance of Best BN Structures from the gold BN Structures with respect to SHD, Recall, Precision

	SHD		Recall		Precision	
	UBCOBaN	*Banjo*	*UBCOBaN*	*Banjo*	*UBCOBaN*	*Banjo*
ASIA	2	2	0.75	0.825	0.814	0.866
ALARM	3	4.4	0.935	0.926	0.955	0.922
HAILFINDER	5	14	0.975	0.927	0.933	0.845
HEPAR II	6	25.4	0.98	0.876	0.963	0.864
ANDES	1	12.8	1.00	0.986	0.997	0.969

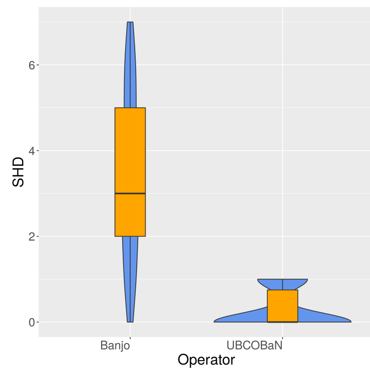
Tables (6.5) and (6.6) show how *UBCOBaN* compares to *Banjo* in terms of their stability measured as the SHD of the returned BN structure from the prior BN Structure. The results shown in the tables are averages of all the 30 runs of incremental application of the Belief Change Operator on the BN structure obtained from the previous run and the data simulated from the gold BN structure. *UBCOBaN* was found to return BN structures closer to the prior BN structures than *Banjo* for all the networks considered. This indicates that *UBCOBaN* was consistently returning almost similar BN structures for each dataset simulated. Thus, *UBCOBaN* was less sensitive to the changes in the samples used to effect Belief Change, as long as the samples are simulated from the BN structure the Belief Change operator aims to rediscover. The *recall* and *precision* values for *UBCOBaN* were higher than those of *Banjo* for all the networks, indicating that *UBCOBaN* was doing better than *Banjo* in returning almost all and only the edges in the prior BN structure. This is evidence that *UBCOBaN* was better at adhering to the principle of *minimal change* than *Banjo*. Figure 6.4 shows the superimposed violin and box plots of the SHDs of the Best BN structures from the prior BN structures. The plots show that the SHD values for *UBCOBaN* were distributed over a narrower range compared to those from *Banjo*, implying that *UBCOBaN* is a more stable Belief Change operator compared to *Banjo*.

Table 6.5.: Average distance of Best BN Structures from the Prior BN Structures with respect to ME, EE, CED, and CE

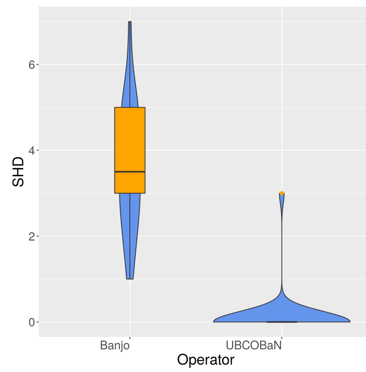
	ME		EE		CED		CE	
	UBCOBaN	*Banjo*	*UBCOBaN*	*Banjo*	*UBCOBaN*	*Banjo*	*UBCOBaN*	*Banjo*
ASIA	0.13	1.167	0.13	1.133	7.8	6.167	7.8	7.267
ALARM	0.03	0.667	0	0.7	44.93	42.77	45	45
HAILFINDER	0.03	0.6	0.03	0.8	69	66.9	69	71.33
HEPAR II	3.2	8.87	3.47	9.0	121	107	121	112
ANDES	0.867	4	0.8	4	338.27	334	338.3	340

Table 6.6.: Average distance of Best BN Structures from the Prior BN Structures with respect to SHD, Recall, Precision

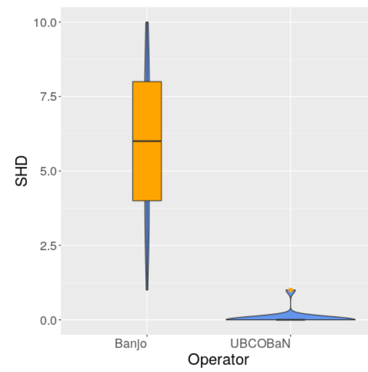
	SHD		Recall		Precision	
	UBCOBaN	*Banjo*	*UBCOBaN*	*Banjo*	*UBCOBaN*	*Banjo*
ASIA	0.267	3.4	0.984	0.75	0.984	0.75
ALARM	0.1	3.6	0.998	0.937	0.999	0.936
HAILFINDER	0.067	5.83	0.999	0.93	0.999	0.927
HEPAR II	6.97	22.83	0.972	0.886	0.970	0.885
ANDES	1.7	14	0.997	0.971	0.998	0.971



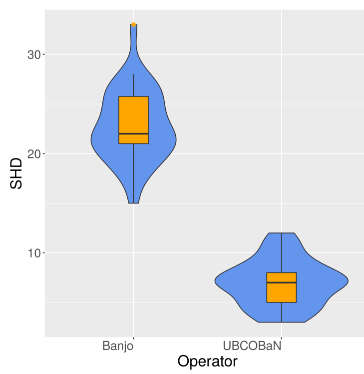
(a) ASIA Network



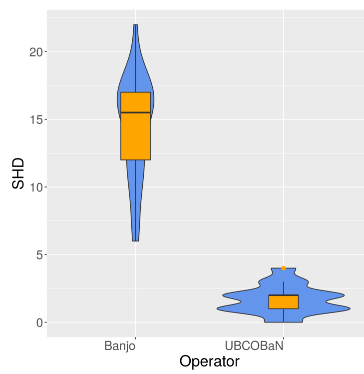
(b) ALARM Network



(c) HAILFINDER Network



(d) HEPAR II Network



(e) ANDES Network

Figure 6.4.: Stability of UBCOBaN based on the dispersion of SHD of the best BN structure from the prior BN structure

6.4.1.4 Experimental Results: The effect of Sample Size on the Accuracy and Stability of the Belief Change Operators

The next set of results report on the effect of sample size of the data used to effect belief change on the *accuracy* and *stability* of the Network structure. In these experiments, data was simulated from the gold BN, and *UBCOBaN* and BANJO were used to effect belief change on the current network structure.

To evaluate the effect of sample size on the accuracy of the operator, the resulting best BN structure and its corresponding Belief Base were compared to the gold BN structure, which was used to simulate the data, and for evaluating stability the resulting network was compared to the prior network. 10 runs were conducted for each sample size. For each run, the network from the previous run was used as the prior network. The ASIA network was not used in these experiments.

The results in Table 6.7 show the comparison of *UBCOBaN* and BANJO in terms of the SHD of the best BN structure returned from the gold BN structure. Generally, the BN structures returned by *UBCOBaN* were closer to the gold BN structures than those returned by *Banjo*. For the ALARM and ANDES networks, *UBCOBaN* returned BN structures close to the gold standard BN structures, even for small sample sizes. The SHDs of the best BN structues returned by *UBCOBaN* from the gold BN structures were not dependent on the sample size. For the HAILFINDER and the HEPER II networks, the SHDs of the best BN structures returned by *UBCOBaN* from the gold BN structures were generally decreasing with increases in the input size. For *Banjo*, the SHDs of the best BN structures returned from the gold BN structures were generally decreasing with increases in the input size for all the benchmark networks considered. These results indicate that *UBCOBaN* is more robust than *Banjo*.

For the HAILFINDER and HEPAR networks, both models struggled to return the correct BN for smaller sample sizes. The poor results obtained for the HAILFINDER and the HEPAR II networks could be due to the parameters in the bn.fit models[6] in the bnlearn package [7] which was used to simulate the data used for belief change in the experiments. The fact that both *Banjo* and *UBCOBaN* obtained better results on ANDES, a larger BN compared to the HAILFINDER and the HEPAR II networks, shows that the poor performance in the HAILFINDER and the HEPAR II network could be due to something particular to the networks.

An interesting result was observed in the application of *UBCOBaN* to the ANDES network for a sample size of 200. An average SHD 0.2 from the gold BN structure of was observed. This is more likely to be due to the entrenchment of the correct edges in

[6]http://www.bnlearn.com/bnrepository/discrete-large.html#hailfinder,
http://www.bnlearn.com/bnrepository/discrete-large.html#hepar2

[7]http://www.bnlearn.com/bnrepository/

the Belief States, which could not be offset by the contribution from the observations. BANJO does not have any mechanism for handling the belief state hence the scores from the sample is all the information it uses to evolve the BN structure. However, this result was not observed for the other networks.

Table 6.7.: The effect of sample size on the accuracy of the operators measured as the SHD of the best BN structures from the gold BN structures

	ALARM		HAILFINDER		HEPAR II		ANDES	
	UBCOBaN	*Banjo*	*UBCOBaN*	*Banjo*	*UBCOBaN*	*Banjo*	*UBCOBaN*	*Banjo*
200	3.2	24.9	44	55.6	73.7	138.5	0.2	188
2200	4.1	6.8	18.6	32.5	48.7	72.1	5.3	59.1
4200	4	7.2	8.9	20.3	32.3	52.7	0.4	35.4
6200	3.9	6	5.6	13	14.3	38.6	1	16
8200	4.1	5.7	4.1	13.3	15.3	32.3	0.3	10.1
10200	3.9	5.8	4.1	10.8	9.4	29.2	1	10
12200	4	5	4	9.1	8.4	25.1	0.9	10.9
14200	3.1	5.1	4.1	13	7.3	23.7	1.3	9.36
16200	3	4.6	3.9	12.4	8.1	22.3	1.1	8.7
18200	3.1	3.8	4.2	9.4	10.5	19.6	1.2	7.8
20200	3	3.3	4	12.7	8.7	20.6	1.3	8.4

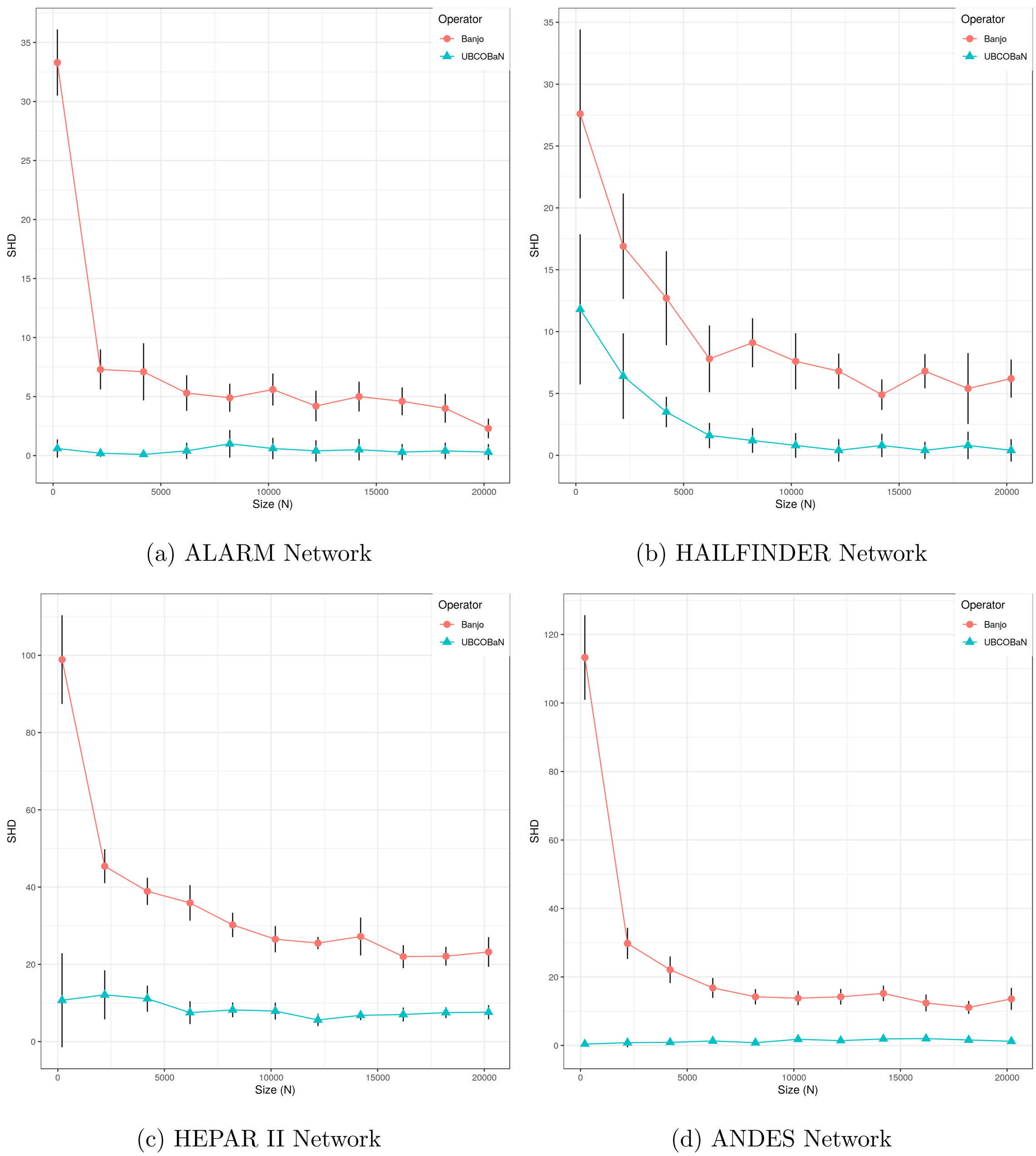
The effect of the sample size on the accuracy of *UBCOBaN* in terms of the Belief Bases returned was also investigated. Table (6.8) shows the effect of the sample size of the data used to effect Belief Change on the Belief Bases returned by the *UBCOBaN* and the *Banjo* operator. The results generally show that the SHD of the Belief Bases from the gold BN structure decreases with increases in the sample size. However, for application of *UBCOBaN* on the ANDES network no well defined pattern was observed.

The Belief Bases from *UBCOBaN* were found to be generally closer to the gold network than those from *Banjo*, except for the HAILFINDER network. The results on HAILFINDER confirm the results which are reported in Table (6.1), that *Banjo* returned Belief Bases closer to the gold BN structure than *UBCOBaN*. This is owing to the same reason given for the results in Table (6.1).

Table 6.8.: The effect of sample sizes on the accuracy of the operators measured as the SHD of the Belief Bases from the gold BN structure

	ALARM		HAILFINDER		HEPAR II		ANDES	
	UBCOBaN	BANJO	*UBCOBaN*	BANJO	*UBCOBaN*	BANJO	*UBCOBaN*	BANJO
200	4.3	17.9	45	54.9	86.3	116.5	0.1	157
2200	5	6.9	40.3	32.7	63.7	82.7	5.4	60.9
4200	4.1	6.4	34.7	24.9	50.4	67.6	0.2	41.7
6200	4	6	28.3	16.7	25.1	52.9	0.1	19.6
8200	4	5.8	25.5	14.7	25	47.9	0.2	14.8
10200	4	6.2	24.2	14.6	19.8	40.9	0.1	13.9
12200	4	5.7	23.4	13.6	17.4	36.3	0.4	14.6
14200	3.6	6.1	23.9	15.8	15.3	34.4	0.7	12.6
16200	3	5.8	23	15.6	14.2	29.9	0.1	13.8
18200	3.1	5.2	22.4	13.9	17.3	29.2	0.4	11.8
20200	3.2	4.9	23.3	15.2	13.6	26.4	0.3	13.2

To evaluate the effect of sample size on the stability of the resulting BN structures, the best BN structures returned were compared to the prior BN structures used in the Belief Change process. Figure (6.5) shows how the BN structures returned by both *UBCOBaN* and *Banjo* compare with the prior BN structure using the average SHD for each sample size. The 95% Confidence Interval (CI) error bars for operators do not overlap. This shows that a Statistical Test of difference between two means will conclude that *UBCOBaN* returned BN Structures that are significantly closer to the prior BN structure than those returned by Banjo for all the sample sizes. This confirms the results shown in Table 6.6 that *UBCOBaN* is generally more stable than *Banjo*. The length of *UBCOBaN*'s 95% CI error bars for the ALARM and the ANDES Networks seem be independent to sample size. However, for the HAILFINDER and HEPAR II network the 95% CI error bars seem to be decreasing with increases in the input size. A comparison of the sizes of *UBCOBaN*'s error bars for each sample size to those from *Banjo* confirms the findings from Figure (6.4) that *UBCOBaN* is generally a more stable Belief Change Operator then *Banjo*.



(a) ALARM Network

(b) HAILFINDER Network

(c) HEPAR II Network

(d) ANDES Network

Figure 6.5.: The effect of sample size on the Stability of the Belief Change Operators

The average SHD of the best BN structures returned by *UBCOBaN* from the prior BN structure was almost constant for all the networks considered except for the HAILFINDER network where the SHD was found to be decreasing with increases in the sample size. For *Banjo* the SHDs were decreasing with increases in the sample sizes for all the networks. This implies that the stability of *Banjo* is dependent on the size of the data used to effect Belief Change.

6.4.2 Evaluating the Agility of the Belief Change Operator

To evaluate the sensitivity of the operator to new information inconsistent with current BN structure, only the ALARM and the ANDES networks were used. For each of the BNs, nine (9) Bayesian Network structures of varying Structural Hamming Distance from the gold BN structure were created. The structure with the least Hamming Distance from the gold standard was labelled Str_1 and the one with the furthest distance was labelled Str_9. The gold standard was labelled Str_0, and Str_2 through to Str_8 were labelled accordingly with respect to their distance from the gold standard. Each structure had its two closest (with respect to the Structural Hamming Distance) neighbours labelled adjacent to it. Before starting the experiments for which the data was recorded, *UBCOBaN* was repeated applied on Str_9 with data simulated from Str_9 up until the structure had converged to a stable Str_9.

The two operators were then used to evolve the BN structure from Str_9 to Str_0. For each structure, Str_i, data to evolve the structure was simulated from $Str\_(i-1)$ and the respective operator would be applied up until the structure had converged or 15 iterations had been reached. The structure from the previous run was used as the prior network, and in moving from one structure to the next one, the structure that was returned from the last run of the previous BN, $Str\_(i-1)$, was used as the prior network. The BN structure used to simulate the data Str_i was used as the *target* BN structure against which the best returned BN structure was compared. The metrics; ME, EE, CED, CE, SHD, recall, and precision, measuring the distance of the returned BN structure from the target BN structure were recorded. Four(4) experimental rounds of evolving the BN structure from Str_9 to Str_0 were carried out for each Belief Change operator. The last 5 runs for each structure from each of the 4 rounds were used to compute the average values of the metrics considered in this study and the standard deviations of the SHD of the BN structures returned by the operators from the target BN structures.

Tables (6.9) to (6.10) show the results obtained for ALARM Network. The results do not show much difference between *UBCOBaN* and *Banjo*, with respect to all the metrics.

Table 6.9.: ALARM:Average distance of Best BN Structures from the target BN Structures with respect to ME, EE, CED, and CE

	ME		EE		CED		CE	
	UBCOBaN	*Banjo*	*UBCOBaN*	*Banjo*	*UBCOBaN*	*Banjo*	*UBCOBaN*	*Banjo*
Str_8	35.4	34.87	0.0667	1.53	47.33	43.67	51.6	49.47
Str_7	23.87	22.27	0	0.67	9.13	12.33	19.13	20.73
Str_6	18.4	16.47	0.0667	0.47	23.4	25.33	27.6	39.53
Str_5	17	16.53	0	1.13	22	19.27	24	24.47
Str_4	20.87	19.8	0	0.8	42	43	44.13	45.2
Str_3	1	1	0	0.6	24.67	23.8	27	27
Str_2	16	15.07	0.0667	0.53	41.67	41.2	44	44.93
Str_1	0.2667	0.27	0	0.87	43.4	41.87	44.73	44.73
Str_0	1	1	0	1.27	42.25	42.27	45	45

Table 6.10.: ALARM: Average distance of Best BN Structures from the target BN Structures with respect to SHD, Recall, and Precision

	SHD		Recall		Precision	
	UBCOBaN	*Banjo*	*UBCOBaN*	*Banjo*	*UBCOBaN*	*Banjo*
Str_8	39.73	42.2	0.57	0.56	0.999	0.97
Str_7	33.87	31.33	0.28	0.36	1	0.95
Str_6	22.67	21.13	0.56	0.61	0.997	0.98
Str_5	19	22.87	0.56	0.54	1	0.94
Str_4	23	22.8	0.67	0.68	1	0.98
Str_3	2.33	4.8	0.96	0.96	1	0.98
Str_2	18.4	19.33	0.72	0.73	0.998	0.99
Str_1	1.6	4	0.99	0.99	1	0.98
Str_0	3.75	5	0.92	0.98	1	0.93

However, a comparison of the standard deviations on the SHD of the best returned BN structure from the target BN structure showed that *UBCOBaN* was more consistent in the best BN structures it was returning than *Banjo*. The standard deviations on the last 5 runs for each structure is shown in Table (6.11). The standard deviation of the SHDs for *UBCOBaN* were all less than 1, whereas those from *Banjo* were greater than 1.3.

Table 6.11.: ALARM: STDEV of the SHD from the *target* BN, calculated from the last 5 of the 10 runs

	STDEV on SHD	
	UBCOBaN	*Banjo*
Str_8	0.96	2.24
Str_7	0.99	2.01
Str_6	0.72	1.85
Str_5	0.00	2.53
Str_4	0.00	1.08
Str_3	0.49	1.18
Str_2	0.63	1.59
Str_1	0.51	1.85
Str_0	0.00	1.31

The results for the ANDES network (See Figures 6.12, 6.13 and 6.14) showed that *UBCOBaN* returned best BN structures that are much closer to the BN structure simulating data than BANJO. Generally the SHD was decreasing as the *target* BN structures got closer to the gold BN structures.

Table 6.12.: ANDES:Average distance of Best BN Structures from the target BN Structures with respect to ME, EE, CED, and CE

	ME		EE		CED		CE	
	UBCOBaN	*Banjo*	*UBCOBaN*	*Banjo*	*UBCOBaN*	*Banjo*	*UBCOBaN*	*Banjo*
Str_8	5.93	11.7	0.8	3.1	132.3	127.05	155.1	149.3
Str_7	4.27	13.4	0.93	4.2	150.33	141	170.73	161.6
Str_6	7.53	24.65	5.4	6.6	198.33	192.15	213.47	204.85
Str_5	5.13	24.95	5.93	9.3	236.67	218.85	249.86	230.05
Str_4	1.67	14.25	4.73	8.6	256.53	243.6	267.33	254.75
Str_3	1.2	7.5	6.86	8.8	270.53	261.95	277.8	271.5
Str_2	1.93	6.46	6.73	9.38	292.47	286.33	298.07	293.54
Str_1	0.33	4.95	5.87	9.45	321.33	313.7	323.67	319.05
Str_0	0	2.3	5.73	10.15	336	331.75	338	335.7

A comparison of the standard deviations on the SHD of the best returned BN structure from the gold BN structure showed that *UBCOBaN* was more consistent in the best BN structures it was returning than BANJO. The average standard devia-

Table 6.13.: ANDES: Average distance of Best BN Structures from the target BN Structures with respect to SHD, Recall, and Precision

	SHD		Recall		Precision	
	UBCOBaN	*Banjo*	*UBCOBaN*	*Banjo*	*UBCOBaN*	*Banjo*
Str_8	29.53	37.05	0.82	0.79	0.84	0.83
Str_7	25.6	38.2	0.86	0.83	0.88	0.87
Str_6	28.07	43.95	0.90	0.84	0.91	0.91
Str_5	24.27	45.45	0.93	0.86	0.93	0.91
Str_4	17.2	34	0.95	0.91	0.94	0.93
Str_3	15.33	25.85	0.97	0.94	0.95	0.93
Str_2	14.27	23.04	0.97	0.95	0.96	0.95
Str_1	8.53	19.75	0.99	0.97	0.98	0.96
Str_0	7.73	16.4	0.99	0.98	0.98	0.96

tions on the last 5 runs for each of the structures is shown in Table (6.14). *UBCOBaN* had less standard deviations on all the other BN structures except for *Str*_6.

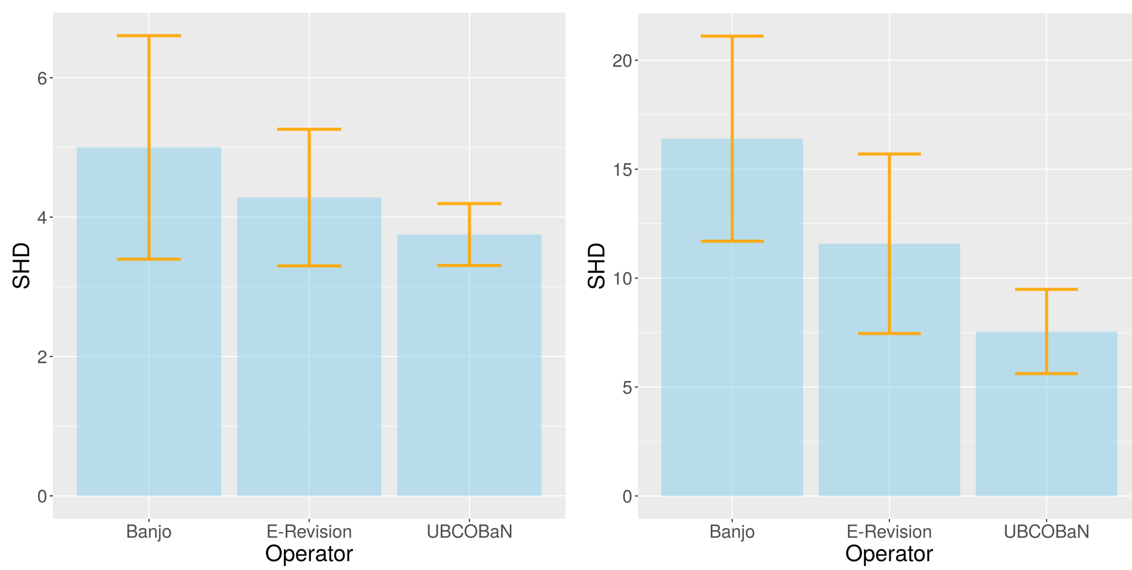
Table 6.14.: ANDES: STDEV of the SHD from the *target* BN, calculated from the last 5 of the 10 runs

	STDEV on SHD	
	UBCOBaN	*Banjo*
Str_8	1.92	4.76
Str_7	2.23	5.71
Str_6	6.24	5.4
Str_5	4.23	5.04
Str_4	3.84	4.24
Str_3	0.9	4.36
Str_2	2.2	3.85
Str_1	2.2	3.18
Str_0	2.22	4.71

6.4.3 Empirical Evaluation of the Importance of Belief Update Component in the Unified Belief Change Model

This section presents an empirical investigation into the importance of the Belief Update component in the Unified Belief Change Model presented in this study. In order to do so, we introduce another operator for Belief Change created by removing the update component from the UBCOBaN operator. For easier reference to the resultant operator, we named it *Epistemic Revision* (*E-Revision*). *E-Revision*, like *UBCOBaN*, uses epistemic states as input to a Belief Change process but it however uses the unupdated Epistemic States.

The experiments carried out in Section 6.4.2 were repeated but this time around with the *E-Revision* operator. The results obtained were combined with the results in Section 6.4.2 for *UBCOBaN* and *Banjo* to obtain the results presented in Figure 6.6. The plots in Figure 6.6 were plotted from the the average of the last 5 runs of each of the four rounds of BN structure evolution from *Str*_9 to *Str*_0. Thus, the results only show the average SHDs of the returned best BN structures from *Str*_1 for *Str*_0 (the gold BN Structure).



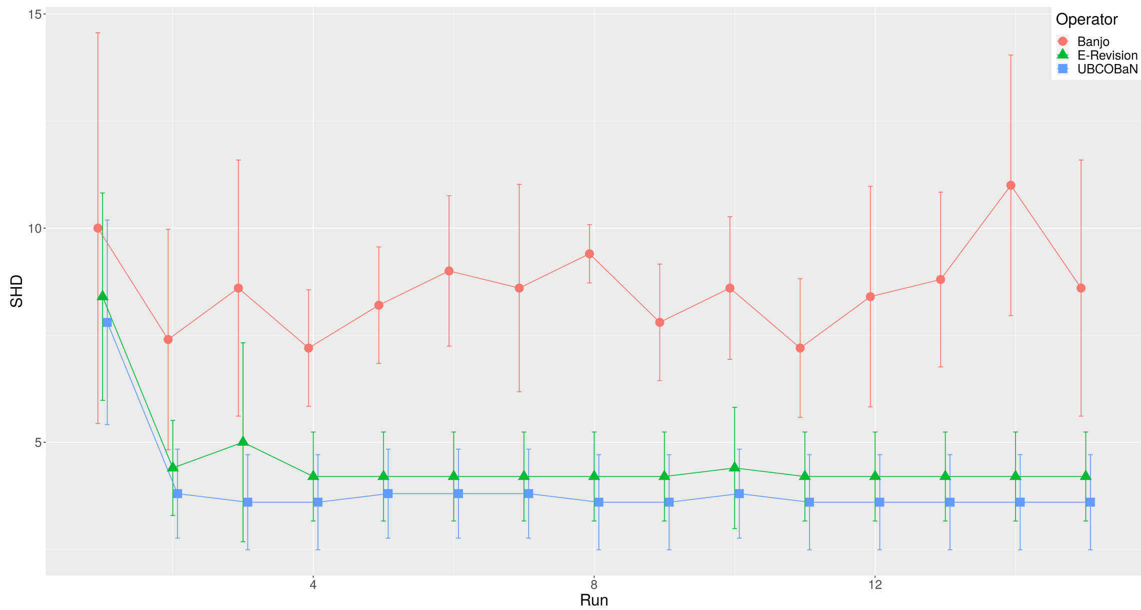
(a) ALARM Network (b) ANDES Network

Figure 6.6.: A Comparison of SHD for *Banjo*, *E-Revision*, *UBCOBaN*

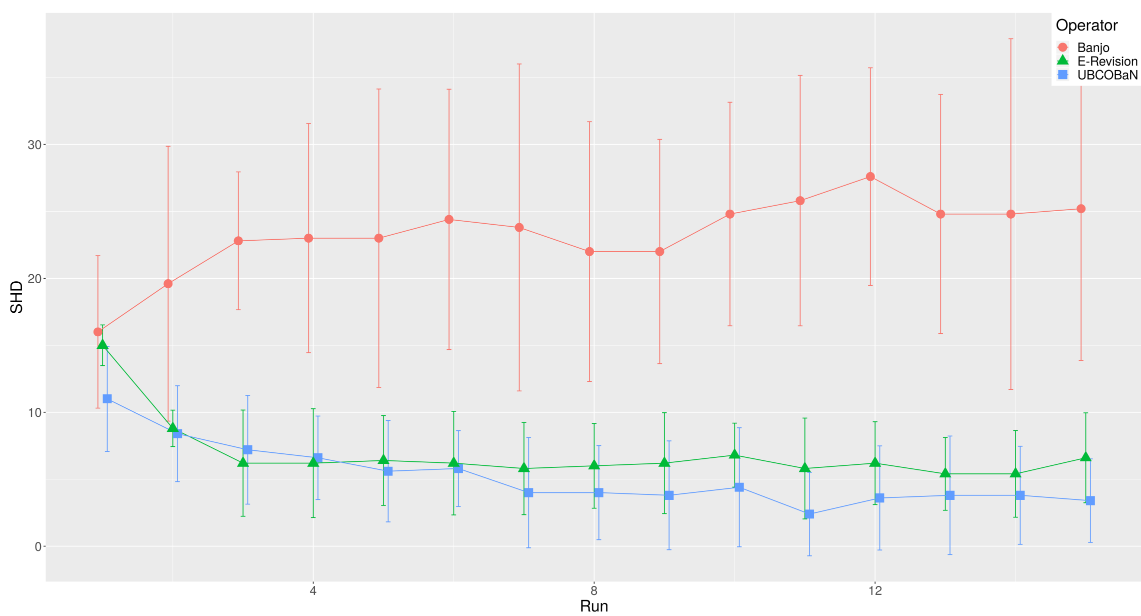
The error bars shown in Figure 6.6 are 95% Confidence Interval (CI) error bars. The results for the ALARM Network show that even though the structures returned by *UBCOBaN* seem to have the least SHD from the gold BN structure followed by *E-Revision* and then *Banjo*, the differences are not necessarily statistically significant. The error bars overlap for all 3 operators. For the ANDES Network the results show that the BN structures returned by *UBCOBaN* are significantly closer to the gold BN structures than the ones returned by *Banjo*. Although the BN structures returned by *E-Revision* seem to be closer to the gold BN structure than those returned by *Banjo* and further than those returned by *UBCOBaN*, the results are also not statistically significant.

The results this far presented show that *UBCOBaN* is a better Belief Change Operator when compared to the classical Search-and-Score algorithms represented by *Banjo*. However, there seems to be no significant difference between *UBCOBaN* and *E-Revision*. In a bid to investigate the importance of the Belief Update component to the Unified Belief Change Model, we set up experiments to investigate the effect of the Update component to the convergence *UBCOBaN* to the true network structure. *Banjo* was repeatedly applied to the BN structure from the previous run, with data simulated from *Str_9* up until the operator was almost consistently returning the same BN structure. Thereafter, the respective operators were applied using the data simulated from the corresponding gold BN structure. For each operator, 5 rounds of of evolving *Str_9* to *Str_0* were carried out, and for each run 15 runs of Belief Change were performed. For the ALARM network 2000 samples were simulated for each run, and 5000 samples were simulated for each run on the ANDES network. The decision on the number of samples to be simulated for each network was influenced by the size the networks.

The aim of these experiments was to investigate the behaviour of each operator as the returned BN structures converge to the gold BN structure from *Str_9*. Figure 6.7 shows the results of convergence analysis of the best BN structures towards the gold BN structure. The results show that the rate of convergence from *Str_9* to the gold BN structure of the ALARM network for *E-Revision* seems to be the same as that of *UBCOBaN*. However, *UBCOBaN* generally returns BN structures closer to the gold BN structure for each run. For the ANDES Network *E-Revision* seems to converge to gold BN structure faster than *UBCOBaN* from run 1 to run 2 but thereafter it slows down. From run 5 *UBCOBaN* began to return BN Structures



(a) ALARM Network



(b) ANDES Network

Figure 6.7.: Convergence of the Belief Change Operators based on the Best BN structures

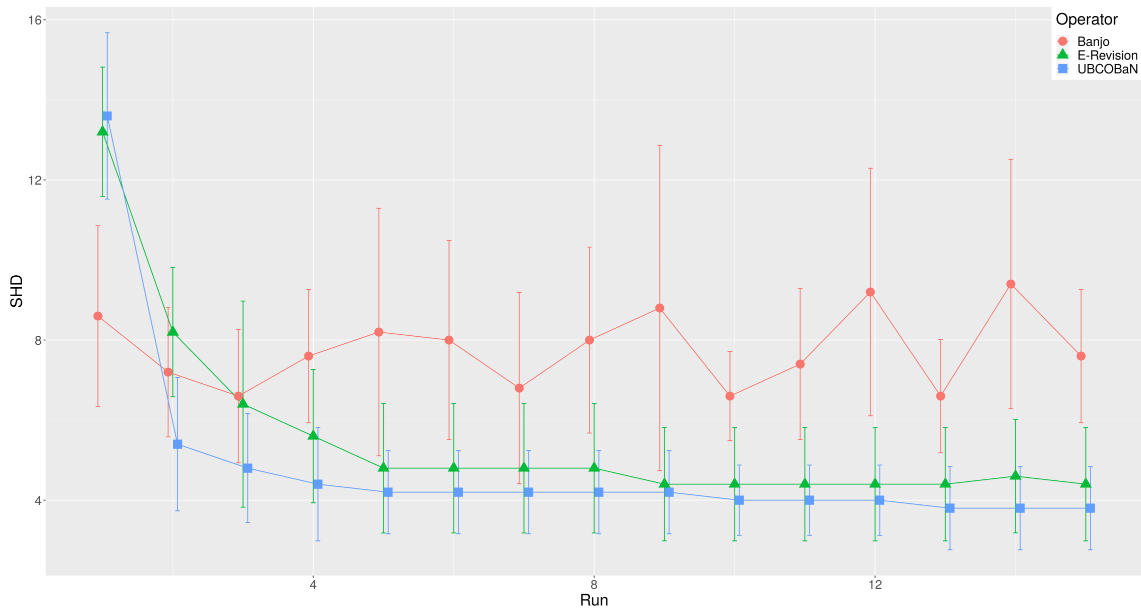
that are structurally closer to the gold BN structure than *E-Revision*. An analysis of the convergence patterns for both networks does not give anything conclusive on which operator, between *UBCOBaN* and *E-Revision* converges faster to the target BN structure. However, the results though not conclusive, show that *UBCOBaN* returns BN structures closer to the gold BN structure than E-Revision for both networks. This seems to support the fact that the Belief Update component plays an important role in the Unified Belief Change Model as hypothesised in the conceptualisation of the Unified Belief Change Model.

We also further investigated the convergence of the Belief Bases towards the gold BN structure. Figure 6.8 presents the results of the analysis. Generally, the Belief Bases from *UBCObaN* seem to converge faster towards the gold BN structure. However, just like the results obtained in Figure 6.7, the differences between the average SHDs of Belief Bases returned by *UBCOBaN* and *E-Revision* from the gold BN structure for each run are not statistically significant, though Belief Bases from *UBCOBaN* seem to be closer to the gold BN structures.

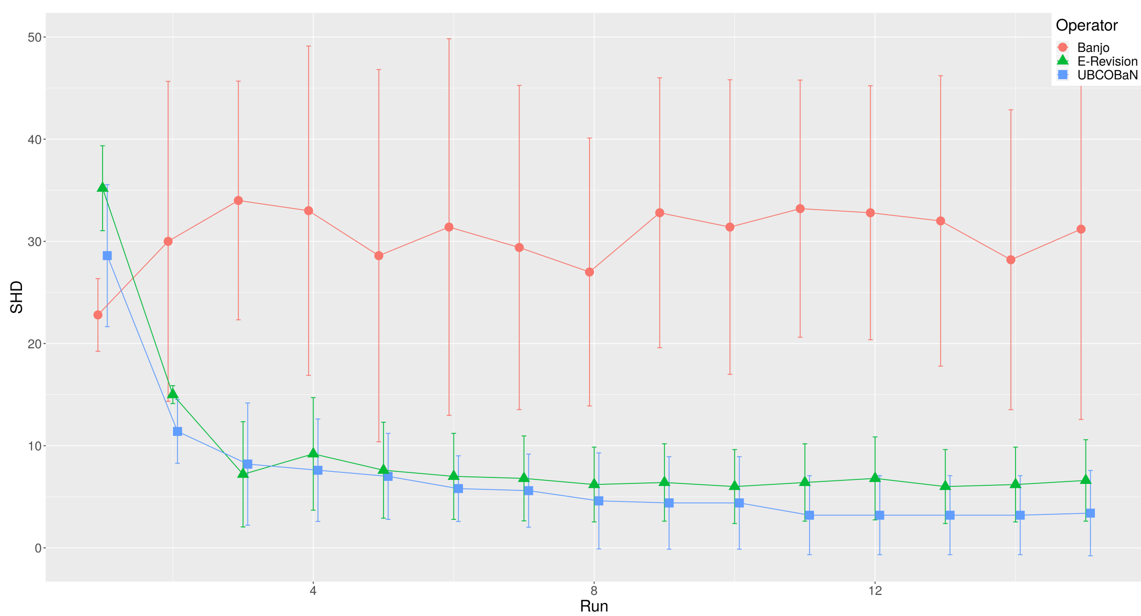
6.5 Discussion of Results

This chapter sought to establish whether the Unified Belief Change Model defined in this thesis adheres to the *principle of minimal change*. This was investigated from both the context of the Belief Bases and the best BN structures returned by the Belief Change Models. *UBCOBaN*, an instance of the Unified Belief Change Model defined in this thesis, was implemented and used to evaluate the model benchmarked on the classical Search-and-Score algorithm implemented in *Banjo*.

The results obtained from the evaluation on Belief Bases showed that *UBCOBaN* is more accurate in returning structural dependences in the process emitting the data than *Banjo* as anticipated from the design of the Unified Belief Change Model discussed in Chapter 4. The Structural Hamming Distance (SHD) of the Belief Bases returned from the gold BN structure was however, higher than expected for the HAILFINDER and the HEPAR II networks given their sizes. A similar pattern was also observed for HAILFINDER and the HEPAR II networks on the SHD of the



(a) ALARM Network



(b) ANDES Network

Figure 6.8.: Convergence of the Belief Change Operators based on the Belief Bases

best BN structures from the gold BN structures. Even though the SHDs for the HAILFINDER and the HEPAR II network obtained in our work were higher than expected compared to the other networks used in our experiments given their sizes, the values seem to be significantly lower than the SHD values obtained for the same networks in related work (Lee & Kim, 2019; Zhao et al., 2015). In the work by Zhao et al. (2015) on using Curriculum Learning for structure learning, the HAILFINDER network had a SHD above 20 for a sample size of 10000 samples and the HEPAR II network had an SHD above 40, compared to SHDs of 4.1 and 9.4 obtained by UBCOBaN for the HAILFINDER and the HEPAR II networks respectively for the same sample size. Lee and Kim (2019) investigated using the BN structure gleaned from a human expert as prior knowledge to be revised using observed data. The results obtained had much higher SHD values than those obtained in our work and the work done by Zhao et al. (2015). For a sample of size 10000, an average SHD above 100 was observed for the HAILFINDER network compared to SHDs of 4.1 and 10.8 obtained in our study for *UBCOBaN* and *Banjo* respectively. The lower SHDs obtained in our results are more likely due to the fact that our experiments used prior BN structures that may be much closer to the target BN structures than the the ones used in (Lee & Kim, 2019; Zhao et al., 2015).

All the results obtained on the Structural Difference of the returned best BN structures from their corresponding prior BN structures showed that *UBCOBaN* returned BN structures closer to the prior BN structures than *Banjo*. Thus, the proposed Unified Belief Change Model was found to adhere to the *principle of minimal change* better than *Banjo*. To the best of the researcher's knowledge, there is no any previous work that has explicitly investigated whether Bayesian Structure learning algorithms adhere to the *principle of minimal change*. Thus, there are no any existing findings in the existing body of knowledge that our findings can be compared to.

Showing that *UBCABaN* adheres to the principle of minimal change is not enough for a Belief Change Model. It could be possible that the models achieve minimal change at the expense of effecting a change where it is necessary. To this end, this study investigated whether *UBCOBaN* is agile enough to change the BN structure when data inconsistent with the current structure is observed. This investigation was done only on the ALARM and ANDES networks.

For the ALARM network not much difference was observed between the agility of *Banjo* and *UBCOBaN*. However, an analysis of the standard deviation of the distances of the the BN structures returned by both *Banjo* and *UBCOBaN* showed that *UBCOBaN* converges faster to a BN structure closer to the target BN structure, and then consistently returned almost the same BN structure for each run.

For the ANDES network, *UBCOBaN* was found to be both more agile and consistent than *Banjo*. *UBCOBaN*'s average SHD from the target BN structure was found to be less than that of *Banjo* for all structures, from *Str*_8 down to *Str*_0. The standard deviations calculated from the last 5 runs for each structure were much smaller compared to those for *Banjo*.

From the foregoing discussion, it can be concluded that *UBCOBaN* adheres to the principle of minimal change better than *Banjo*, at least in propositional Bayesian Networks. It can also be concluded that it does so without compromising its agility to change the BN structure when evidence not consistent with the current BN structure is observed. The agility of *UBCOBaN* can be attributed to the Epistemic States and the Belief Update component. Experiments carried out with a version of *UBCOBaN* without the Belief Update component, *E-Revision*, showed that use of Epistemic States significantly improves the rate of convergence of the returned BN structures towards the target BN structures as the number of Belief Change iterations increase. A comparison of the convergence of the BN structures from *E-Revision* to those of BN structures from *UBCOBaN* did not show any statistically significant difference between the two, but BN structures returned by *UBCOBaN* tended to converge much closer to the target BN structure as the number of Belief Change iterations increase. This serves as evidence that the update component is important to the Unified Belief Change Model in order for the model to return BN structures that are closer to the true model that is emitting the data. The update component of the model anticipates the most likely transitions from one BN structure in the Belief State to another, and uses this anticipation to estimate the propensities of an edge being added, removed or reversed for any hypothesised BN structure in the Belief State.

Previous work on theory refinement in Bayesian Networks is mainly restricted to revising the BN structure. None of the works reviewed consider Belief Update. Notable works towards theory refinement in Bayesian Networks include (Buntine, 1991; Friedman & Goldszmidt, 1997; Lam & Bacchus, 1994; Liu et al., 2018; Yu, 2019; Yue et al., 2015). Results obtained from the work by Friedman and Goldszmidt 1997 showed that keeping a set of high scoring Bayesian Networks as the epistemic state to inform how the BN structure should be refined gives reasonably good BN structures. Such a result is in agreement with the results obtained in this chapter. The ability of a Search-and-Score algorithm that uses Epistemic States without Update to perform better than the classical Search-and-Score algorithm was due to its use of epistemic states.

The work by Yu (2019) follows a completely different approach to Belief Revision. Instead of keeping a Belief State over high scoring BN structures, the solution keeps the data from the previous iteration of belief revision. The work defined an Adaptive tendency factor, ν, that would be used to determine how much of the overall score of the BN structures should be contributed by new data. if $\nu > 0.5$, the fitness between current Bayesian Network and old data has a larger proportion in scoring function, so the learning process trends to old data. if $\nu < 0.5$, the fitness between the current Bayesian Network and new data has a larger proportion in the scoring function, so the learning process trends to new data. The results obtained showed that the proposed solution outperforms the solution proposed by Friedman and Goldszmidt 1997. This is expected since the solution is in principle similar to the solution referred to as the naive approach in (Friedman & Goldszmidt, 1997). Owing to the fact the solution uses all the data that has been seen so far, it tends to give more optimal BN structures. However, as stated in (Friedman & Goldszmidt, 1997), the solution requires vast amounts of memory to store the entire corpus data. From a Belief

Change perspective such a model will have a strong tendency of persisting beliefs that are no longer acceptable longer than necessary.

The work reported in (Liu et al., 2018) presents a solution for incremental learning of BN structures that only revises the substructures of the the current BN structure that are inconsistent with the observed data. The work defined an influence degree score that measures the variation of BN's probability parameters with respect to the likelihood of the new data. Only the sub-structures within the Markov blanket of the nodes with high influence degrees would be revised. Ideally such a solution should be able to find most of the conditional independences implied by the original BN structure. The evaluation of the approach was done on the ASIA network and the error rate on inference using the learned model was used as the evaluation metric. The results obtained showed that the proposed solution can effectively be used for revising BN structure in relevant applications. However, the study did not investigate how the proposed solution will work in medium to large BN structures.

One of the major deviations *UBCOBaN* has from the classical structure learning algorithm is the use of modular priors for the hypothesised BN structures. As mentioned in (Eggeling et al., 2019), even though a lot of work on Search-and-Score structure learning highlights the importance of the structure prior component in the score metrics, there are hardly any empirical studies that use the component with the exception of (Eggeling et al., 2019; Talvitie, Eggeling, & Koivisto, 2018). Empirical work on structure learning assumes uniform priors. The assumption is the effect of the prior in the overall BN score diminishes with increases in the sample size. *UBCOBaN* uses modular structures to compute the prior for any given BN structure. As shown in the result obtained in (Eggeling et al., 2019) structure learning algorithms that use modular priors return BN structures closer to the ground truth structures than those that assume uniform priors. The use of modular priors was key to our proposed model because it enabled the use of the Belief Update model proposed in this study. Our results also show that use of modular priors performs better than using uniform priors.

The next chapter, Chapter 7, investigates whether *UBCOBaN* can be used for belief change in First Order Probabilistic Logic (FOPL). Multi-Entity Bayesian Networks, and instances of FOPL will be used for this evaluation. The evaluation process to be followed will be the same as the one used in this chapter.

7. BELIEF CHANGE IN MULTI-ENTITY BAYESIAN NETWORKS

7.1 Introduction

The motivation to study Belief Change in the structure of Bayesian Networks in this thesis emanated from the need for rational evolution of First Order Probabilistic Knowledge representations. This chapter discusses how the developed Belief Change Model was used for evolving structure in Multi-Entity Bayesian Networks (MEBN) (Laskey, 2008). The chapter starts off by giving an overview of MEBN with respect to how it conforms to the class of FOPL, and its peculiarities. This is followed by a characterisation of the Structure Learning Problem in MEBN and how *UBCOBaN* was used for evolving the MEBN structure. Lastly, the chapter discusses evaluation of *UBCOBaN* for structure learning in MEBN and the findings thereof.

7.2 Overview of Multi-Entity Bayesian Networks

Multi-Entity Bayesian Networks (MEBN), like Bayesian Networks represent joint probability distributions for a collection of interrelated random variables. The joint probability distributions are captured by a graphical model, with nodes representing uncertain hypotheses about the entities in the domain, and the edges representing probabilistic dependencies between the hypotheses. However, propositional Bayesian Networks have limited expressiveness to represent generic relationships between entities. MEBN extends Bayesian Networks with First Order semantics to represent knowledge about groups of entities and to reason about such knowledge.

Knowledge in MEBN is represented in a flexible modular way as a set of MEBN Fragments, known as MFrags. A set of coherent MFrags is known as an an MEBN

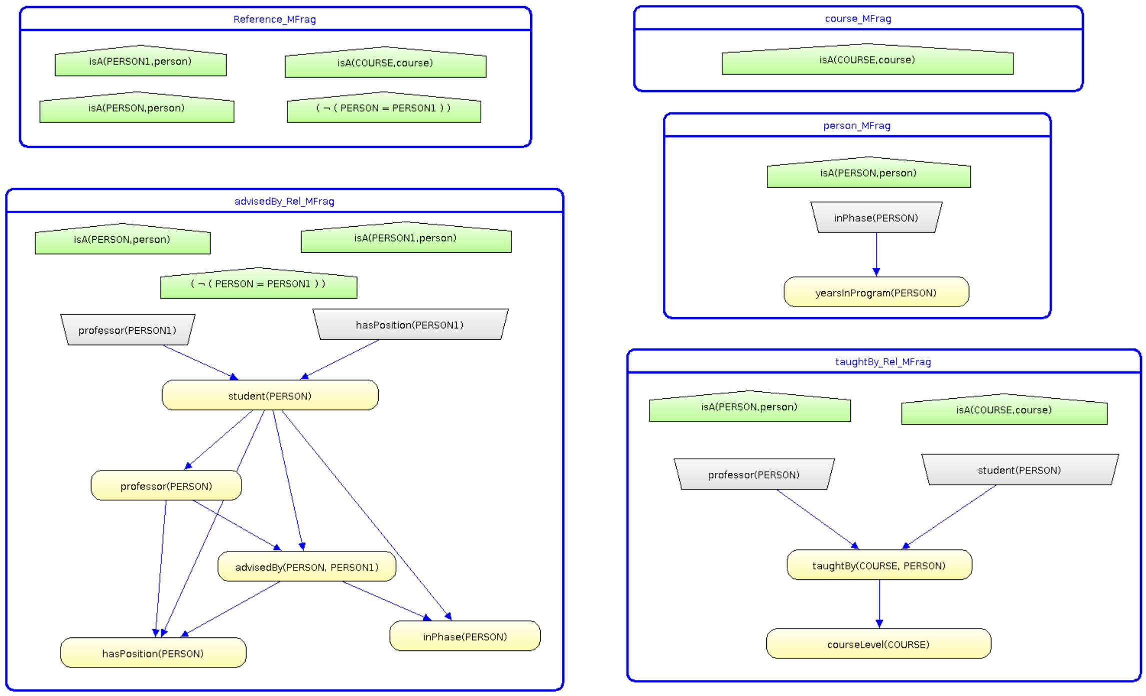


Figure 7.1.: An Example MTheory

Theory (MTheory). Each MFrag represents probability information about a set of related random variables. The random variables are equivalent to First Order Logic predicates or functions. They take arguments that refer to entities in the domain. Each random variable in the MTheory must have exactly one (1) MFrag as its home MFrag. The home MFrag for a random variable is the MFrag were its local distribution is defined.

An MFrag can have three (3) types of random variables. These are *Resident*, *Input* and *Context* random variables. Resident nodes of an MFrag encode the probability distribution local to the MFrag. The Input random variables correspond to the root nodes in a given MFrag whose local distribution is defined in some other MFrag in the MTheory. The context nodes represent the conditions that must be satisfied for the dependencies and the local distribution represented in the MFrag to hold. Figure 7.1 shows an Example MTheory for the UW_std relational dataset. Figure 7.2 shows the legend for the different types of nodes defined in MEBN.

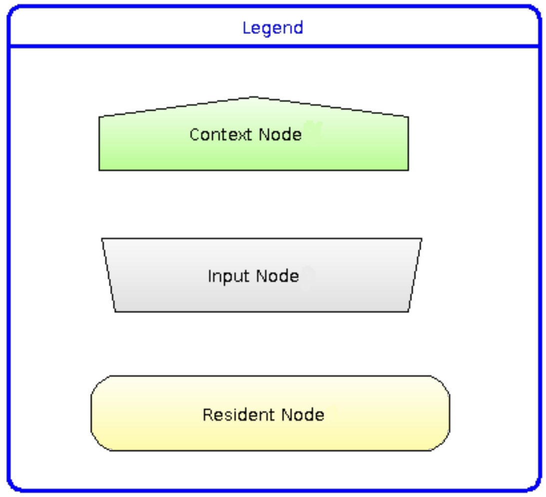


Figure 7.2.: Legend for MEBN Nodes

7.3 The Belief Change Problem in MEBN

MEBNs, like Bayesian Networks have two (2) major components that are important for knowledge representation namely, the graphical structure encoding dependency structure between variables, and the probabilistic regularities characterising uncertainty inherent in the domain. A Belief Change Operator for MEBN should therefore seek to rationally evolve both the structure and the parameters for the MEBN model. The Unified Belief Change Meta-Model defined in this thesis is generic enough to handle both aspects of Belief Change in MEBN. However, this work only focuses on rational Belief Change of the network structure.

In MEBN, just like in Bayesian Networks, there is no guarantee that the statistical regularities of the random variables will be represented by one unique structure. It is possible that, several arrangements of the random variables into the MEBN theory could be encoding the same statistical regularities. This study did not consider equivalences of the different arrangements of the variables in the algorithm for evolving MEBN structure defined in this work. Efforts to address the equivalences in the different MEBN structures was left as part of future extension of the work reported in this thesis. Belief change as defined by the algorithm defined in this thesis seeks to rationally evolve the arrangement of the variables in an MEBN theory and their dependency structure into a structure that best explains variability in the observation and the effect of the changes that could have possibly occurred in the domain. This is exactly the goal of the Unified Belief Change Model for Bayesian Network Structures defined in Chapter 4.

7.4 Using *UBCOBaN* in MEBN

Learning and evolving a BN Network Structure in First-Order Probabilistic Graphical Models can not be easily done from flat data. Classical Bayesian Network structure learning works with “flat“ data representations. Such data does not capture the relational structure that exists between the entities in the domain. The approach to the structure learning problem in MEBN followed in this thesis is inspired by the work presented in (Getoor, Friedman, et al., 2001; Park, Laskey, Costa, et al., 2013). The relational schema is taken as background knowledge. The relational structure of the schema is assumed to be static and there is no need for it to be revised and/or updated. The Relational schema specifies all the entity types that have been observed this far and the relationships between them. The MEBN-RM model defined in (Park, Laskey, Costa, et al., 2013) that suggests how MEBN elements can be mapped to a Relational Model (RM) is adopted.

7.4.1 The MEBN-RN Model

Park, Laskey, Costa, et al. (2013) defined an MEBN-RM bridge for matching MEBN elements to Relational Model elements. MEBN-RN model defined in (Park, Laskey, Costa, et al., 2013), defines 4 types of context nodes; the Isa, $Slot-Filler$, $Value-Constraint$ and the $Entity-Constrant$ context nodes. The adapted model used in this thesis defined only one type of context nodes, the Isa type. The entity table in RM is mapped to the Isa context node variable. It is important to note that a relationship table, whose primary key is composed of foreign keys does not correspond to an Isa random variable. In this thesis, instead of mapping such a relationship table to a context random variable as done in (Park, Laskey, Costa, et al., 2013), the relationship table was mapped to a predicate resident random variable. The $value - constraint$ context and slot-filler context node type were not defined.

In MEBN, resident nodes can be described as a Function or Predicate. Predicates in MEBN are Boolean random variables with possible values $True$ and $False$. The MEBN_RM bridge used in this study maps attributes in a given entity or relationship table to a function, and or Relationship tables to predicates. The arguments to a predicate random variable are the entity types whose keys are making up the primary key for the relationship table. The arguments to a function random variable is the entity type for which the attribute is defined. Both predicates and function are probabilistic and hence they should have a Conditional Probability distribution associated with them.

7.4.2 Belief Change in MEBN

Inspiration for the proposed Belief Change algorithm for MEBN was drawn from the work on structure learning in MEBN done by Park, Laskey, Costa, et al. (2013) . The main difference between the structure refinement algorithm defined in this thesis and the work by Park, Laskey, Costa, et al. (2013) is on the nature of the flattened data used to refine the BN structure and how the structure learnt from relational tables is handled. The discussion below shows how the algorithm defined in our work works and how it differs from that of related work.

The algorithm starts off by identifying all the entity tables and Relation tables in the database using the primary and foreign key structure of the tables. This will be used to instantiate an MFrag for each table seen in the database. A default MFrag is created to hold all the random variables that will be used as context nodes in the entity and relationship MFrags. The relationship MFrags do not only have random variables from its corresponding table, but it will have data from entity tables and other relationship tables it has relational links with. Thus, the data for learning relationship MFrags is flattened data from quite a number of tables. Owing to the fact that all the data in the entity tables may also be contained in one or more data matrices for the relation tables, the researcher gave higher priority to the dependencies learnt from relationship tables. Thus, a random variable will only be made a resident random variable in an MFrag associated with its Entity Table if it does not have a parent in any relationship MFrags. The algorithm for MEBN structure evolution is shown in Algorithm 7.1.

7.5 Experiments

This section provides the evidence that the proposed Unified Belief Change Model and its corresponding Operator can be used for Belief Change in the structure of a MEBN model. To evaluate the performance of the proposed Belief Change model the CORA, WebKP, UW_std, and the Financial_std benchmark Statistical Relational Learning (SRL) datasets were used. First, the research aimed to establish that the *UBCOBaN* Operator does not change the underlying network structure unnecessarily even in the case of relational data. This investigation is similar to the one conducted with propositional data except for the fact that, this time around relational data is used. Second, the study aimed to establish that if the process emitting the data being observed changes the developed Operator will accordingly evolve the structure to reflect the process emitting the data. This is also similar to the process conducted in the previous chapter to prove the agility of the model.

For all experiments that needed data simulation, the *bnlearn*[1] package was used for data simulation. The *Rserve* package[2] was used to enable the *java* program to communicate with *R* for data simulation.

[1]http://www.bnlearn.com/

[2]https://cran.r-project.org/web/packages/Rserve/index.html

Algorithm 7.1 MEBN Structure Learning Algorithm

Input: DB, $maxSlotChain$, $BNSL\_alg$
Output: M_{theory}
procedure BSL_MEBN
    $M_{theory} \leftarrow$ create a default MTheory
    $MF_{ref} \leftarrow$ create a default reference MFrag
    $EntityTables \leftarrow$ create a list of all entity tables
    for each $table \in EntityTables$ **do**
        $entMF_i \leftarrow$ create an empty MFrag for the table
        $entMF_i \leftarrow$ add an Ordinary RV for the entity
    $relationshipTables \leftarrow$ create a list of all relationship tables
    $JTList \leftarrow$ JOINTABLES($DB, relationshipTables, maxSlotChain$)
    for each $JT \in JTList$ **do**
        $MF_{rel} \leftarrow$ create an empty relationshipMFrag
        $relG_k \leftarrow$ effect belief change for relationship MFrag using $BNSL\_alg$ and $data_k$
        $relMF_k \leftarrow$ CONVERTDAGTOMFRAG($relG_k, relMF_k, M_{Theory}$)
    for each each Entity MFrag $entMF_i \in M_{Theory}$ **do**
        $entRS_i \leftarrow$ create a resultset from the remaining RVs in $entMF_i$
        $data_i \leftarrow$ persist the resultset into a data file
        $entG_i \leftarrow$ effect belief change for the Entity MFrag DAG using $BNSL\_alg$ and $data_i$
        $entMF_i \leftarrow$ CONVERTDAGTOMFRAG($entG, entMF_i, M_{Theory}$)
    for each $MF_i \in M_{Theory}$ **do**
        $MF_i \leftarrow$ CALCULATEPARAMETERS(MF_i, DB, M_{Theory})
procedure JOINTABLES(DB, RELATIONSHIPTABLES, MAXSLOTCHAIN)
    for $k = 0$ untill relationshipTables.size() **do**
        $frgnKeysList \leftarrow$ add all foreign keys in relationshipTable
        $relMF_k \leftarrow$ create an empty relationship MFrag
        $MF_{ref} \leftarrow$ create an OV for each foreign table add it to MF_{ref}
        $relMF_i \leftarrow$ add OVs to $relMF_i$
        $RV_{pred} \leftarrow$ create a predicate RV using OVs as arguments

for $s = 1$ untill maxSlotChain **do**
$JT_i \leftarrow$ join all related tables in the range maxSlotChain
$RS_k \leftarrow$ create a resultset from the joint tables
$Data_k \leftarrow$ persist the resultset in a data file
$JTList.add(JT)$
procedure CONVERTDAGTOMFRAG(DAG, MF, M_{Theory})
if $MF \in relationshipTables$ **then**
for each $node \in$ DAG **do**
if $node$ is an RV in MF **then**
$parentList \leftarrow$ add all parent nodes of node
for each $parentNode \in$ parentList **do**
$node.add(parrentNode)$
for each $node \in MF$ **do**
if $node.getParents == 0$ && $node.getChildren == 0$ **then**
if node is aa attibute in some entity table **then**
release the node to the entity MFrag
else
for each $node \in$ DAG **do**
if $node$ is an RV in MF **then**
$parentList \leftarrow$ add all parent nodes of node
for each $parentNode \in$ parentList **do**
$node.add(parrentNode)$
procedure CALCULATEPARAMETERS(M_{Theory}, DB)
for each $MF_i \in M_{Theory}$ **do**
$SQLStatement \leftarrow$ create an SQL statement for all RVs in the MFrag
$data_i \leftarrow$ get data from DB using the SQl statement
$MF_i \leftarrow$ learn parameters from $data_i$

7.5.1 Benchmark Statistical Relational Learning Datasets

The benchmark relational datasets used in this set of experiments were categorised as small, medium sized, and large dataset. The categorisation was done on the basis of the number of relational tables that a dataset has. The WebKP dataset (Craven et al., 1998) was originally constructed for learning computer understandable First-Order Logic knowledge for the World Wide Web. The dataset consisted of three (3) relational tables: webpage, content, and cites. The webpage table has webpage URLs as the primary key and class of a webpage as the only attribute for the table. The content table has two attributes the *webpage_id* (as the primary key) and *word_cited_id*. The cites table holds the citation network of the webpages. The citation network consists of 1608 links. The CORA dataset is closely related to the WebKP dataset, but instead of using webpages, it uses a citation network of scientific research articles. The purpose of the CORA dataset (McCallum, 2017) was to learn FOL knowledge for citation link prediction and multi-label classification prediction. On the Medium-sized dataset, the UW-CSE dataset [3] was used. The UW-CSE dataset describes an academic department (15 predicates; 1323 constants; 2673 ground atoms). This dataset lists facts about the Department of Computer Science and Engineering at the University of Washington (UW-CSE). The main prediction problem for the dataset is determining who is the advisor of who. The database has four(4) tables; person, course, advisedBy, and taughtBy. Some related works that have used this dataset include (Dinh, Vrain, & Exbrayat, 2012; França, Zaverucha, & d'Avila Garcez, 2014). The Financial_std dataset was used in the category of large datasets. The dataset was used for the PKDD'99 Challenge. The dataset has eight(8) tables holding data on 606 successful and unsuccessful loans.The target for the Financial_std dataset is to predict successful loans.

The SRL datasets are different from the benchmark propositional Bayesian Learning datasets, in the sense that the true network/FOL statements are not known. The

[3] https://relational.fit.cvut.cz/dataset/UW-CSE

datasets are usually used to evaluate Inductive Logic Programming (ILP) solutions based on how good the learned Inductive logic rules are at predicting unknown classes. This will not be possible in our case since this thesis did not provide a solution for learning the Conditional Probability tables for the learnt MEBN models. Since the parameter learning is outside the scope of this thesis, further efforts in evaluating the developed belief change model in MEBN through inferences will be considered in future research efforts. Further to the foregoing, for the Belief Change problem in FOPL, it is more appropriate to use Structural Distance than prediction accuracy. This is due to the fact that it is possible for a FOPL model far away from the true model (in terms of structural distance) to give more accurate prediction than models much closer to the true model. To this the end, the evaluation process followed in evaluating propositional Bayesian Networks was used. Owing to the fact that the ground truth Relational BN structures are unknown, accuracy of the returned MEBN models based on the edges returned could not be evaluated. Hence only stability of the algorithms was evaluated.

7.5.2 Evaluating the Stability of the Unified Belief Change Model on FOPL

The experiments conducted here were meant to investigate the stability the Best FOPL Bayesian Network structure. The results obtained from using *UBCOBaN* were benchmarked against the results obtained from using *Banjo* to effect Belief Change.

The first set of experiments on evaluating the stability of the Belief Change Operators was done as follows: First, a MEBN Theory was learnt from the original benchmark relational datasets. This was necessary because the benchmark relational datasets were originally meant for Inductive logic Programming and hence no First Order probabilistic graphical model is provided. The dataset also does not give the FOL statement from the dataset. The MEBN structures learnt from the data using the MEBN structure learning algorithm proposed in this thesis are shown in Figures 7.3 - 7.6

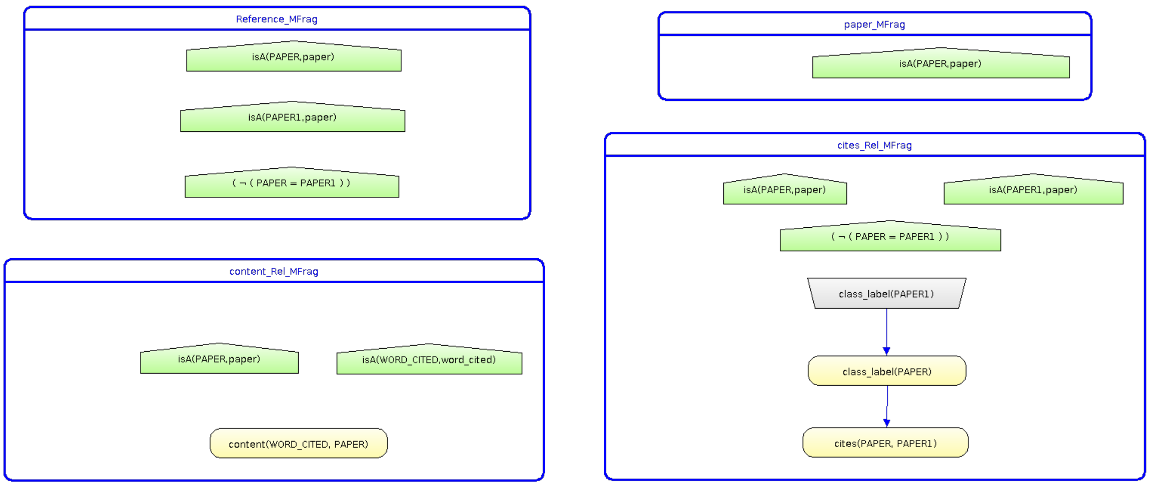


Figure 7.3.: CORA MTheory

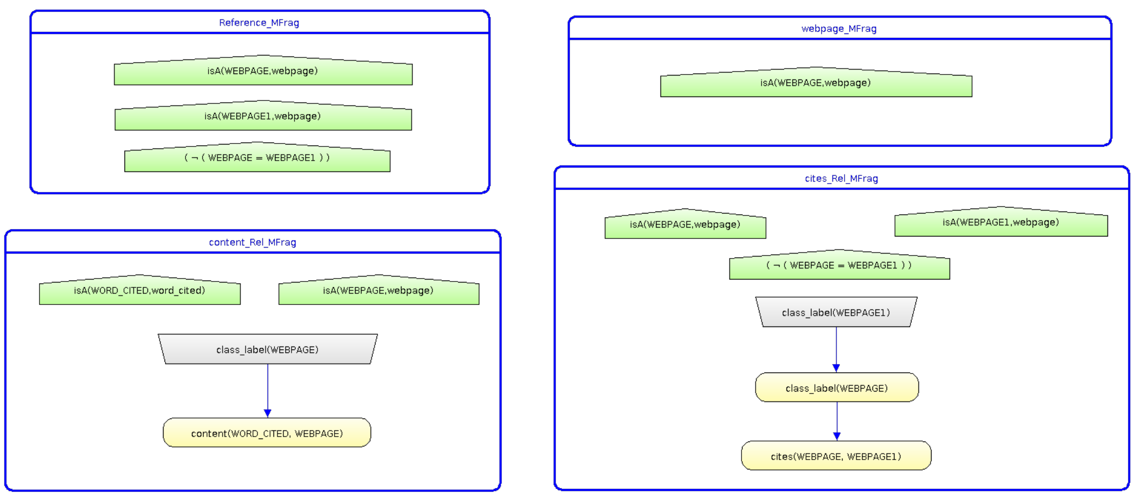


Figure 7.4.: WebKP MTheory

The metrics used for evaluation were ME, EE, CED, ICED, SHD, Recall and Precision. However, these metrics were measured comparing the current MTheory to the prior MTheory. Stability of the Belief Change solution is evidenced by a very low SHD and very high Recall and Precision.

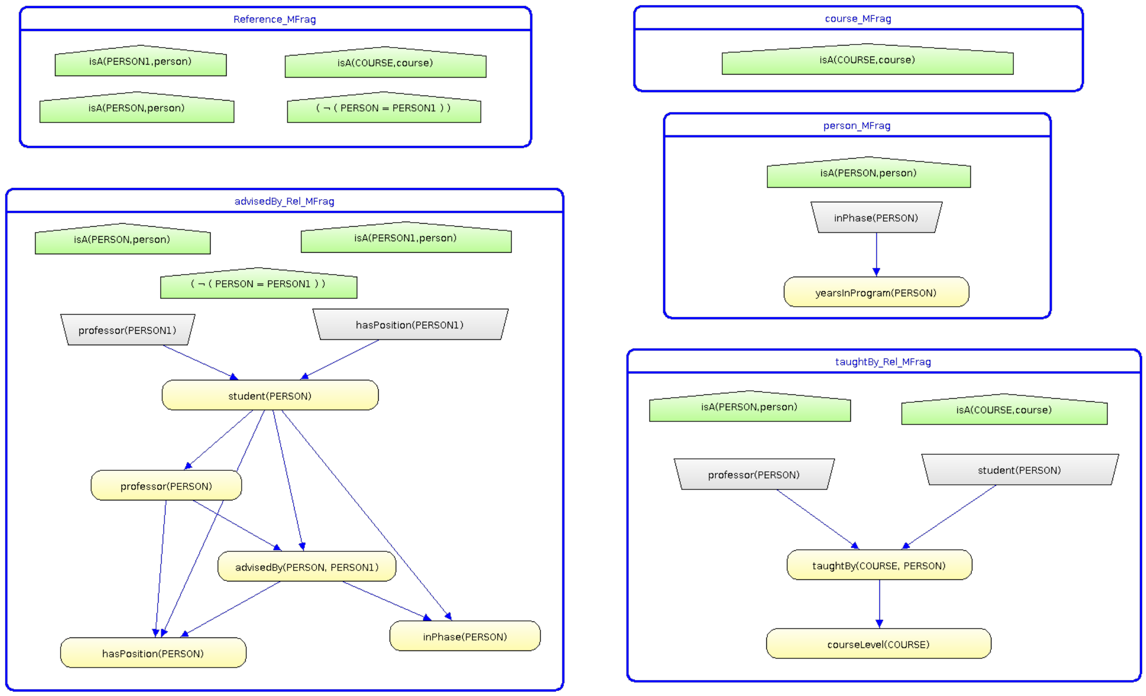


Figure 7.5.: UW_std MTheory

For each MTheory, data was simulated using the Bayesian Network from each of the MFrags in the Mtheory. 30 runs of Belief Change on the MTheory were conducted, using the MTheory from the previous run as the Knowledge representation on which Belief Change will be effected. 2000 samples were simulated for each run and the structure search algorithm was run for a maximum of 2 minutes. The choice of 2 minutes was made after observing that owing to the small size of the MFrag DAGs, the search process hardly gets to 2 minutes before exhausting all the possible graph structures. The maximum number of DAGs to be returned by the searcher was set to 100. However, the number of DAGs found was hardly reaching 100. Table (7.1) below shows the results obtained. The values recorded in Table (7.1) reflect the average of the last 5 runs of incremental application of Belief Change from the MTheories shown in Figures (7.3) to (7.6)

In all the relational schemas used in this study, *UBCOBaN* was found to return MTheories that are closer to the prior network than *Banjo*. The differences in the

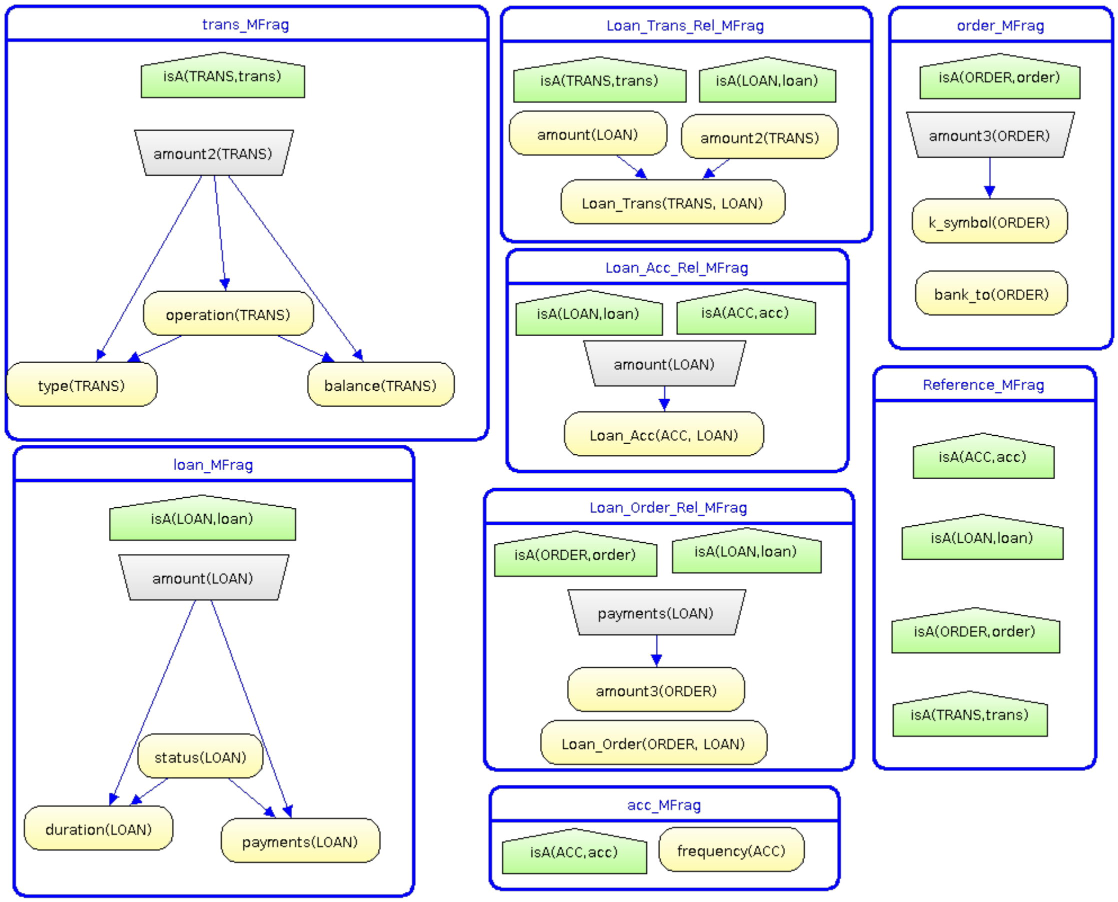


Figure 7.6.: Financial_std MTheory

SHDs from the prior network between the two models was more defined for large MTHeories, UW_std and Financial_std. The results show that, *UBCOBaN* is more stubborn than *Banjo* in terms of changing the dependence structure of the MFrags in the MTheory.

For UW_std, *UBCOBaN* deviated from the prior MTheory by 0.6 SHD units compared to 7.2 SHD units for *Banjo*. For Financial_std, *UBCOBaN* deviated from the prior MTheory by 1.2 SHD units compared to 5 units for *Banjo*. The SHD, Recall and Precision metrics show that *UBCOBaN* was found to be more stable and adhered to the principle of minimal change much better than *Banjo*.

Table 7.1.: Stability of the Operators based on the difference between the Best MTheory and the prior MTheory: ME, EE, CED, CE

	ME		EE		CED		CE	
	UBCOBaN	*Banjo*	*UBCOBaN*	*Banjo*	*UBCOBaN*	*Banjo*	*UBCOBaN*	*Banjo*
CORA	0	0.2	0	0.2	2	2	2	2
WebKP	0	0.2	0	0.2	3	2.6	3	2.6
UW_std	0.2	3.4	0.4	3.8	14	9.8	14	9.8
Financial_std	0.8	2.6	0.8	2.4	12.2	13.6	12.2	13.6

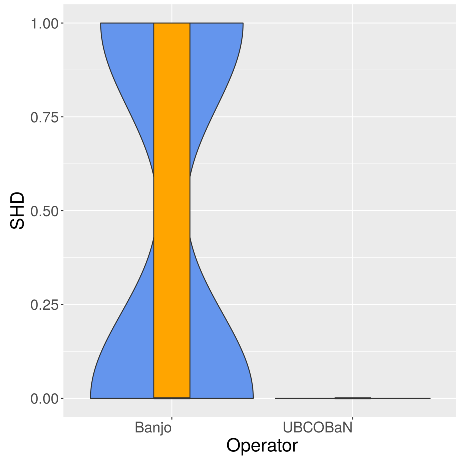
Table 7.2.: Stability of the Operators based on the difference between the Best MTheory and the prior MTheory: SHD, Recall, Precision

	SHD		Recall		Precision	
	UBCOBaN	*Banjo*	*UBCOBaN*	*Banjo*	*UBCOBaN*	*Banjo*
CORA	0	0.4	1	0.93	1	0.93
WebKP	0	0.4	1	0.93	1	0.93
UW_std	0.6	7.2	0.98	0.75	0.97	0.73
Financial_std	1.6	5	0.916	0.85	0.913	0.86

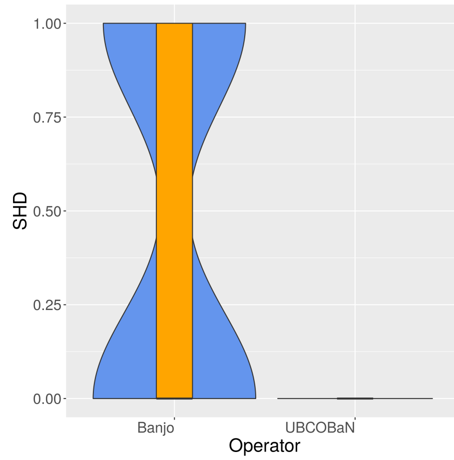
Figure 7.7 shows the distribution of the SHDs of the best returned MTheories from the prior MTheories. The results show that *UBCOBaN* consitently returned the same MTheory for the small relatioanl shemas, CORA and WebKP. For the *UW_std* and the *Financial_std* schemas, the SHDs had much lesser dispersion compared to those from *Banjo*. This implies that *UBCOBaN* is a more stable Belief Change Operator compared to *Banjo*.

7.5.3 Evaluating the Agility of the Belief Change Meta-Model on FOPL

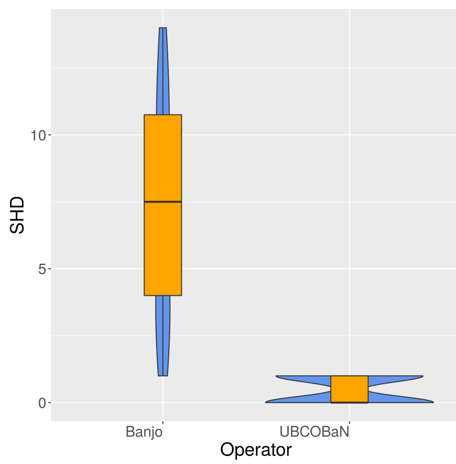
To evaluate agility of the *UBCOBaN* in refining the MTheory relative to changes in the domain, only the *UW_std* and the *Financial_std* datasets were considered. The MTheories from the WebKP and the CORA datasets were too small to be considered for these experiments.



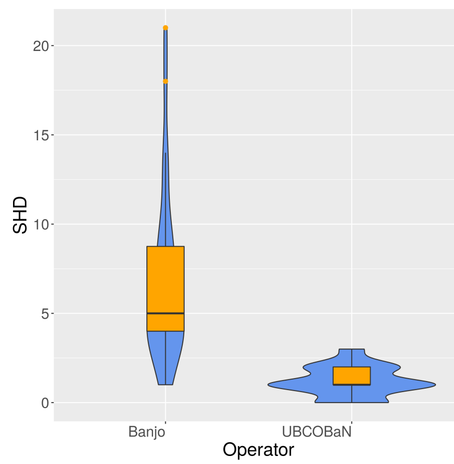
(a) CORA MTheory



(b) WebKP MTheory



(c) UW_std MTheory



(d) Financial_std MTheory

Figure 7.7.: Stability of Best MTheory Returned

Owing to the small sizes of the DAGs in the MFrags only three (3) DAGs were synthetically created for each MFrag. The DAGs were characterised as follows:

Str_0 was the DAG learnt from the benchmark dataset. *Banjo* was used as the structure learning algorithm. *Str_2* was a DAG with no edges between any of its nodes. *Str_1* was a DAG with some edges in *Str_0*, but the total number of edges was less than the number of edges in *Str_0* and some of the edges in *Str_1* were not in *Str_0*. Only three (3) MTheories were created for each database schema owing to the small size of DAG in the MFrags. The DAGs for most of the MFrags were too small to allow any latitude for creating more than 3 different DAGs. This is expected for FOPL owing to the fact that the DAG for FOPL are much smaller than those from their propositional counterparts.

For each MTheory, from $MTheory\_2$ to $MTheory\_0$, data was simulated using $MTheory\_i$ and 15 runs of incremental refinement of the MTheory would be carried out. The MTheory from the previous run would be used as the prior MTheory that would be refined using *UBCOBaN* or *Banjo* and the data simulated from the target MTheory. The target MTheory could be $MTheory\_1$ or $MTheory\_0$. The first run of the 15 runs for $MTheory\_i$ would use the result from the last run from $MTheory\_(i-1)$, as its initial structure. 2000 samples were simulated for each run, and the search time for each MFrag in the MTheory was set to 2 minutes. Four (4) experimental rounds of evolving the MTheories from $MTheory\_2$ to $Mtheory\_0$ were conducted for each Operator. The metrics ME, EE, CE, CED, ICED CE, SHD, Recall and Precision, measured against the target MTheory, were recorded. The aggregated results are shown in Tables (7.3) to (7.6).

The results obtained for the *UW_std* schema shows that *UBCOBaN* returned MTheories that are closer to the target MTheories than *Banjo*. On $MTheory\_1$ *UBCOBaN* had an average SHD of 5 units from the target MTheory whereas *Banjo* had an SHD 6.67 units. On $MTheory\_0$ *UBCOBaN* had an average of 0 SHD from the prior MTheories. An Investigation into how fast the Belief Change solution converges to the MTheories simulating the data on *UW_std* showed that *UBCOBaN* converges faster than *Banjo* (see Figure 7.8). Figure 7.8 shows the analysis done on $MTheory\_0$. By run 13 *UBCOBaN* began to consistently return the MTheory simulating the data whereas *Banjo* was still struggling to consistently return MTheories close to the MTheory simulating the data.

The results obtained for the *Financial_std* MTheory also showed better agility for *UBCOBaN*. The MTheories returned by *UBCOBaN* were found to have a lower SHD from the target Mtheory than those returned by *Banjo* for both $MTheory\_1$ and

Table 7.3.: *UW_std*:Comparison of *UBCOBaN* against *Banjo* based ME, EE, CED, and CE

	ME		EE		CED		CE	
	UBCOBaN	*Banjo*	*UBCOBaN*	*Banjo*	*UBCOBaN*	*Banjo*	*UBCOBaN*	*Banjo*
MTheory_1	2.47	3.93	2.53	4.13	11	9.73	11	9.73
MTheory_0	0.27	1.53	0.27	1.47	13.07	12.53	13.07	12.53

Table 7.4.: *UW_std*: Comparison of *UBCOBaN* against *Banjo* based SHD, Recall, and Precision

	SHD		stdev(SHD)		Recall		Precision	
	UBCOBaN	*Banjo*	*UBCOBaN*	*Banjo*	*UBCOBaN*	*Banjo*	*UBCOBaN*	*Banjo*
MTheory_1	5	8.07	0.09	4.62	0.82	0.73	0.82	0.73
MTheory_0	0.53	3	1.4	3.53	0.98	0.89	0.98	0.89

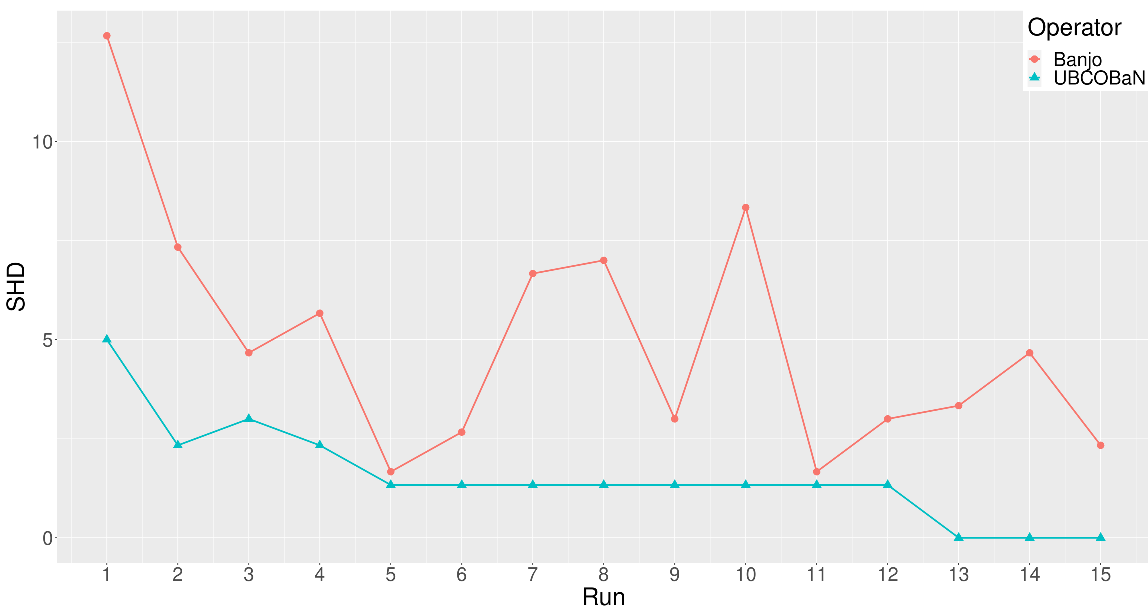


Figure 7.8.: *UW_std*: Convergence of the Evolving MTheory to the MTheory simulating the data

MTheory_0. The standard deviations of the SHDs of the last 5 runs of the 15 runs per MTheory showed that *UBCOBaN* consistently returned MTheories that are closer to the target MTheories better than *Banjo* did. Figure 7.9 shows the convergence trends of the average SHD as the experiments move from the 1^{st} run towards the 15^{th} for $MTheory_0$. The figure shows *UBCOBaN* converges faster towards the MTheory simulating the data than *Banjo*.

Table 7.5.: *Financial_std*: Comparison of *UBCOBaN* against *Banjo* based on ME, EE, CED, and CE

	ME		EE		CED		CE	
	UBCOBaN	*Banjo*	*UBCOBaN*	*Banjo*	*UBCOBaN*	*Banjo*	*UBCOBaN*	*Banjo*
MTheory_1	0.93	4.2	0.8	4.2	8.27	9.07	8.27	9.07
MTheory_0	0.4	4.53	0.73	4.4	13.6	9.8	13.6	9.8

Table 7.6.: *Financial_std*: Comparison of *UBCOBaN* against *Banjo* based on SHD, Recall, and Precision

	SHD		stdev(SHD)		Recall		Precision	
	UBCOBaN	*Banjo*	*UBCOBaN*	*Banjo*	*UBCOBaN*	*Banjo*	*UBCOBaN*	*Banjo*
MTheory_1	1.73	8.13	2.31	4.6	0.89	0.76	0.900	0.76
MTheory_0	1.13	8.93	1.51	5.76	0.97	0.70	0.94	0.71

7.6 Discussion of Results

The experimental results presented in Section 7.5 show that *UBCOBaN* adheres to the principle of minimal change better than the classical Search-and-Score algorithms implemented in *Banjo*. The results were in the affirmative for all the benchmark relational schemas considered in this study. For small relational schemas, CORA and WebKP, *UBCOBaN* was faithful to the principle of minimal change so much that none of the 30 runs conducted for each schema returned an MTheory deviating from the prior MTheory. For the large relational schemas, *UW_std* and *Financial_std*, *UBCOBaN* returned MTheories that slightly deviated from the the prior MTheory at average SHDs of 0.6 and 1.2 for *UW_std* and the *Financial_std* respectively. This is much less than the SHDs for the MTheories returned by *Banjo* which were 7.2 and 5 for *UW_std* and *Financial_std* respectively.

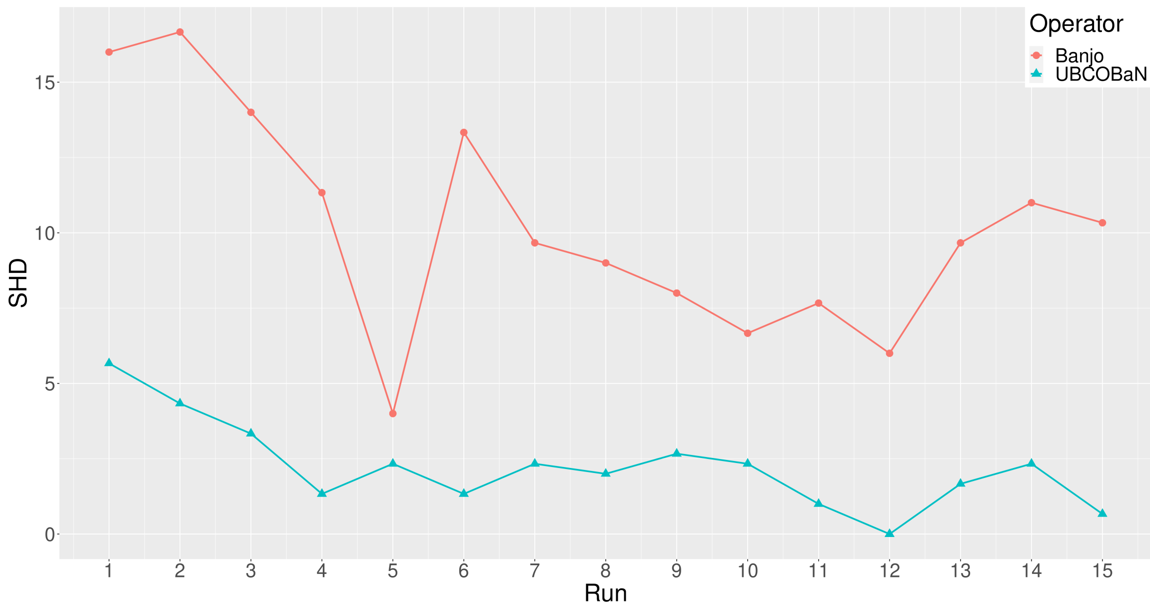


Figure 7.9.: *Financial_std*: Convergence of the Evolving MTheory to the MTheory simulating the data

The results obtained from the experiments for evaluating the agility of *UBCOBaN* in MEBN showed that *UBCOBaN* is more agile than *Banjo*. The average SHDs of MTheories returned by *UBCOBaN* were closer to the target Mtheories than those returned by the classical Search-and-Score algorithm implemented in *Banjo*.

There is not much work that has been done on theory refinement in MEBN except for the work done by Park *et al* in (Park & Laskey, 2018; Park, Laskey, Costa, et al., 2014, 2016; Park, Laskey, da Costa, et al., 2013), which considers the problem from a Structure Learning perspective. The work presented in this thesis builds upon these works. Our work has two (2) major deviations from Park *et al* (2014): (i) our model models Existence Uncertainity (Getoor, Friedman, et al., 2002); and (ii) our MEBN structure learning algorithm gives precedence to learning relational structure over learning entity structure.

As highlighted in Section 1.3 of this thesis, no work has explicitly investigated Belief Change in Probabilistic Graphical Models. Like most of the work on Structure Learning in Statistical Relational Learning, the focus of the work by Park, Laskey, Costa, et al. (2014) is on learning structure rather than Belief Change in Structure. Thus, the work does not assume existence of any structure priors, let alone the use of Epistemic States in the Structure Learning process. Other related work on Structure Learning in Statistical Relational Learning such as (Coutant, Leray, & Le Capitaine, 2014; Ettouzi, Leray, & Messaoud, 2016; Friedman, Getoor, et al., 1999; Getoor, Friedman, et al., 2001, 2002; Popescul & Ungar, 2003) also do not assume any prior knowledge on the structure of the First Order Probabilistic knowledge representation. All the solutions investigated assume that the structure is being learnt from scratch.

Early work on Structure Learning in FOPL only considered attribute uncertainty. That is, the uncertainty modelled in the FOPL was pertaining to the values of the attributes. Further studies (e.g. Getoor, Friedman, et al., 2002) on structure learning discovered that modelling structural uncertainty in FOPL improves the prediction accuracy of the First Order Probabilistic Models. The work by Getoor, Friedman, et al. (2002) proposed two types of structural uncertainty representation in Relational data; *Reference uncertainty* and *Existence Uncertainty.* Reference Uncertainty models the process by which reference slots are selected from a given set, and Existence Uncertainty models whether a relationship exists between any two objects. Owing to the fact the results obtained in (Getoor, Friedman, et al., 2002) showed that a model with Existence uncertainty seems to have better predictive accuracy than the one with reference uncertainty on the WebKP dataset, our proposed Belief Change Model models Existence Uncertainty. Due to the fact that we wanted to capture Existence Uncertainty, the proposed model learnt DAG in the Relationship Tables first before learning the DAG for entity tables. Only variables that have not found space in the relationship tables were made resident variables in the entity table MFrags.

Although evaluating the the Belief Change algorithm using predictive accuracy of the refined theories is important to any work on structure learning in FOPL, it does not however, enable one to evaluate how good a belief change algorithm is based on Belief Change principles. It is possible that a model with a very large structural distance from the true model may give better predictive accuracy than a model that is close to the true model. Thus, owing to the fact that we are interested in Belief Change, Structural Hamming Distance was chosen to be the primary metric for evaluating the proposed Belief Change model in MEBN.

8. SUMMARY, CONCLUSIONS, AND FUTURE WORK

This chapter summarises the discourse that was presented in this thesis, draws some conclusions about the intuition and conceptualisation used to develop a Unified Belief Change Model for dynamic computing environments based on the experiments carried out in this study. It also highlights some possible extension of the work presented in this thesis and other contemporary research that can be enriched by some of the ideas developed in this thesis.

8.1 Summary

This thesis sought to address the problem of automatic evolution of Knowledge representations for handling knowledge in Open and highly dynamic computing environments. Such a problem is both ontological and epistemological. This thesis argued that to address such a problem there is a need for a Knowledge Representation framework that inherently handles uncertainty, which is pervasive in Open and Dynamic Computing Environments. The thesis takes the position that FOPL is an ideal KR approach to address the ontological aspects of the problem, and then goes on to provide a solution for the epistemological aspect of the problem that pre-supposes a FOPL KR framework.

Though a lot of research efforts in FOPL assume that the structural knowledge comes from a Knowledge expert, techniques that have emanated from the field of Statistical Relational Learning have the potential to provide solutions to addressing the epistemological aspects of the problem. Evolution of Knowledge Representations in classical logic has been widely studied under the auspices of Belief Change. This thesis therefore extended the techniques that have emanated from the research in

Belief Change in Cassical Logic and Statistical Relational Learning to come up with a solution for Belief Change in FOPL.

Belief Change in Open and Dynamic Computing Environments requires that beliefs be changed in response to: (i) correction of mistaken beliefs about the domain; and (ii) changes in the domain. In classical Belief Change, Belief Change resulting from correction of mistaken beliefs is known as Belief Revision and AGM theory is the most popular solution to the problem. Belief Change due to changes in the domain is known as Belief Update. This thesis, developed a Unified Belief Change Model that caters for both Belief Revision and Belief Update. Inspiration for the proposed Unified Belief Change Model was drawn from the Unified Belief Change solution by Boutilier (1998).

The Belief Change Model was conceptualised by first modelling the evolving Bayesian Network structure as a dynamical system whose impetus for change is driven by the occurrence of some events in the domain. The event semantics were used to capture the Belief Update aspects of the model and conditinalisataion was used to cater for Belief Revision. To ensure iterative Belief Change the model was defined in such way that the prior knowledge used as input to a belief change process was an Epistemic State capturing relative entrenchments of the BN structures and the relative propensities of each and every edge in a given BN structure being given up. The outcome of a Belief Change process is also an Epistemic State holding posterior entrenchments of the BN structures and the propensities of the edges being given up given some BN structure. The derived Belief Update model component of the Unified Belief Change Model was formally validated by analogy using the Qualitative Belief Change Model for Dynamic environments and theory of Partially Observable Markov Decision Processes (POMDP). For the Belief Revision component, Baysian Conditionalisation was used with the outcome of the Belief Update component being used as the prior Epistemic State. It was also proven that the proposed Unified Belief Change Model meets the postulates for revision of p-functions presented in (Boutilier, 1995).

Apart from arguing the efficacy of the proposed belief change model from a theoretical standpoint, this thesis also provides empirical evidence for the same. A Belief Change Operator, the Unified Belief Change Operator for Bayesian Networks (*UB-COBaN*), based on the proposed Unified Belief Change Model was developed. The operator was then used to illustrate how the model achieves belief change using a synthetic example with one (1) iteration of Belief Change. Further to the foregoing, the Operator was implemented in *java* to enable empirical evaluation of the efficacy of the model in both Propositional Bayesian Networks and in Multi-Entity Bayesian Networks (MEBN). MEBN is a variant of FOPL we chose to use for evaluating the proposed model for belief change in First-Order Probabilistic Knowledge Representation. The results obtained show that the proposed model adheres to the principle of minimal change (principle information economy) better than if the classical Search-and-Score algorithms were to be used to effect belief change both in propositional Bayesian Networks and MEBN. The model was also found to be at least as agile as the classical Search-and-Score algorithm for both propositional Bayesian Networks and MEBN in instances where data inconsistent with the assumed network structure is observed. A statistical test on whether Belief Update significantly improves rationality of the proposed Unified Belief Change Model on propositional Bayesian Networks was inconclusive at 95% confidence interval. However, the results obtained showed that the Unified Belief Change Model with Belief Update has superior performance compared to the one without Belief Update, albeit not statistically significant.

8.2 Conclusions

This thesis presented the following arguments for addressing the problem of automatic evolution of Knowledge Representations in Open and Dynamic Computing Environments:

1. There is a need for a KR framework that inherently deals with the epistemological aspects of automatic evolution of KRs,

2. The automatic evolution of KRs problem can be modelled as a Belief Change problem and it needs to cater for both Belief Revision and Belief Update,

3. Being Bayesian about Belief Change in KRs is one way of ensuring rational evolution of Knowledge Representations.

The need for a Knowledge Representation framework that inherently deals with epistemological aspects of automatic evolution of Knowledge Representation is one one of the key deviations of the work presented in this thesis from other related work on evolution of Knowledge Representations. Most work in this area are based on evolution of OWL-based Knowledge Representations, commonly known as ontologies. These representations have no mechanism for handling knowledge acquisition. Such work treat the Knowledge Acquisition and the Knowledge Representation as separate processes. Typically, Description Logic (DL) is used as the knowledge Representation language and the ontology evolution process is thought of as a reconfiguration-design problem (Stojanovic et al., 2003). With such a solution, automatic evolution of the Knowledge Representation becomes impossible, owing to the knowledge acquisition process being very time consuming and needing human intervention. This research argued that First Order Probabilistic Logic is a natural candidate for Knowledge Representation, if automatic evolution of Knowledge Representations is to be achieved. This thesis provides evidence that FOPL based Knowledge Representation can be automatically evolved using techniques from the field of Belief Change and Statistical Relational Learning (SRL).

The solution for evolution of Knowledge Representations presented in this study was based on the techniques that have emanated from the field of Belief Change in Classical Logic. The experiments conducted showed that both Belief Update and Revision are necessary for Belief Change. The Belief Update component of the solution enabled the proposed Belief Change model to return BN structures that are much closer to the true BN structure when data inconsistent with the prior BN structure is observed than the *E-Revision* Operator and the classical Search-and-Score algorithm.

The Belief Change solution presented in this thesis also builds on the ideas that have emanated from Bayesian Inferences. Bayesian Inferences assume that any complex system is dynamic and the structure and the parameters of a model needed to explain variability in the domain are merely models and not necessarily the fundamental truth. Hence in Bayesian Inference the structure and the parameters can be learnt in the form of probability distribution. By doing so, they can be kept flexible and can be updated whenever new information is observed. This thesis provided evidence that it is possible to be Bayesian about Belief Change in structure in Bayesian Networks, and the key requirement like in any Bayesian Inferences environment is being able to define a Belief State as a probability distribution over all possible explanations/models. To ensure Iterative Belief Change in the graphical structure it is necessary that an Epistemic State be used as input and that the result of a Belief Change process should also be an Epistemic State. Using Epistemic States for Belief Change resulted in a Belief Change Operator that consistently returns almost the same graphical structure if data consistent with the prior structure is observed. The inability of classical Search-and-Score structure learning algorithms to use epistemic states in the belief change process resulted in them struggling with consistently returning BN structures that are close to the target BN structure.

The problem that this thesis addressed, can be thought of as part of the grand problem of enabling computers to do Science. This is often referred to in literature as the automatic theory refinement problem. The problem simply put is defined as: given some domain theory and a set of observations, find an approximately minimal set of necessary changes to the domain theory that results in the theory being able to correctly explain variability in the observation (Ourston & Mooney, 1994). This thesis postulated that First Order Probabilistic Logic is a natural choice for a representational formalism for automatic theory refinement. FOPL inherently handles the epistemological needs of theory refinement and does not assume infallibility of the propositions postulated by the domain theory. Thus, the conceptual framework for automatic evolution of knowledge representations presented in this thesis is one effort towards a theoretical framework for enabling computers to do science.

8.3 Future Work

Several promising areas of future research came out of this study. This section discusses some of the limitations the Unified Belief Change Model for Bayesian Networks presented in this study and the possible extension that can be made on the model in the future.

First, the current model was defined to deal with Belief Change in Structure alone and not the parameters. The choice to look only at structure was deliberate owing to the following reasons: (i) the parameter learning and by extension the parameter refinement is assumed to be simpler and well studied problem; (ii) from a knowledge representation perspective the parameters are thought to be not ontological but a mere epistemological convenience for knowledge acquisition and inferences. However, the researcher believes that for a complete theory refinement in Bayesian Network, there is a need for it to be done also on the parameter, and the structure will only be refined when the observations are now inconsistent with the structure. Future research studies should therefore investigate when theory refinement at structure level should be done. Having a parameter learning algorithm for FOPL will open up quite a number of future research opportunities for the proposed model, which are discussed in the next 3 paragraphs.

Since the datasets used for evaluation of the MTheories returned by *UBCOBaN* are benchmark datasets for Inductive Logic Programming, whose evaluation is often done through inferences, it would be desirable to do the evaluation through inferences. This would have enabled *UBCOBaN* to be compared to the techniques that have emanated from ILP in learning First Order Theories. However, such an evaluation requires that the model parameters for the the MEBN model be defined.

Some research works on Bayesian Structure Learning use the Kallback-Leibler(KL) divergence and other related entropy-based metrics to evaluate how close the learned structure is from the gold standard Bayesian Network Structure. The KL-divergence can also be used to measure how much information is lost in moving from one distribution to the other. On the other hand the Kullback-Leibler has also been used to evaluate the principle of minimal change in classical Belief Change work. It would be a worth while research endeavour to investigate how entropy-based metrics, like the Kullback-Leibler divergence, can be used both for ascertaining the principle of minimal change and the evaluate the correctness of the evolved network, both in propositional Bayesian Networks and First Order Probabilistic Logic. However, use of such metrics will require that both the structure and the parameter of the model be estimated. In future, the researcher will explore extension of the Belief Change Model defined in this thesis to handle Belief Change in parameters of a Bayesian Network and use of entropy-based structural difference metrics for evaluation.

Second, this thesis made an assumption that there is no missing data in the sense that for any sample used for Belief Change each and every variable has a value against it. Unfortunately, real-life data is usually incomplete. The major difficulty with incomplete data is computational in nature. The missing data will now need to be estimated, using patterns learned from other samples with complete data. This implies that the samples are nolonger independent and the DBe metric can no longer be given by the sum of the local terms. Thus, further research still needs to be conducted on how the developed Belief Change Model can work with incomplete data.

Third, the model was only defined for Directed FOPL. Literature on FOPL has shown that undirected FOPLs are generic enough to represent all the knowledge that can be represented by their directed counterparts. By extension of the above argument, a theory refinement solution defined for undirected FOPLs should be generic enough to handle Belief Change in directed FOPLs. However dealing with undirected models will add a lot of complexities to the Belief Change problem.

Bibliography

Abramson, B., Brown, J., Edwards, W., Murphy, A., & Winkler, R. L. (1996). Hailfinder: A bayesian system for forecasting severe weather. *International Journal of Forecasting*, *12*(1), 57–71. https://doi.org/https://doi.org/10.1016/0169-2070(95)00664-8. (Last accessed on 24/06/2020)

Alchourrón, C. E., Gardenfors, P., & Makinson, D. (1985). On the Logic of Theory Change: Partial Meet Contraction and Revision Functions. In *Readings in Formal Epistemology*. Springer. https://doi.org/10.1007/978-3-319-20451-2_13. (Last accessed on 24/06/2020)

Alchourrón, C. E., Gärdenfors, P., & Makinson, D. (1985). On the Logic of Theory Change: Partial Meet Contraction and Revision Functions. *J. Symb. Log.*, *50*(2), 510–530.

Antoniou, G., & van Harmelen, F. (2004). Web Ontology Language: OWL. In *Handbook on Ontologies* (pp. 67–92). Berlin, Heidelberg.

Beinlich, I. A., Suermondt, H. J., Chavez, R. M., & Cooper, G. F. (1989). The alarm monitoring system: A case study with two probabilistic inference techniques for belief networks (J. Hunter, J. Cookson, & J. Wyatt, Eds.). In J. Hunter, J. Cookson, & J. Wyatt (Eds.), *Aime 89*, Berlin, Heidelberg, Springer Berlin Heidelberg.

Benferhat, S. (2010). Interventions and belief change in possibilistic graphical models. *ARTIFICIAL INTELLIGENCE -AMSTERDAM- ELSEVIER-*, (2), 177–189.

Ben-Hur, A., Elisseeff, A., & Guyon, I. (2002). A stability based method for discovering structure in clustered data. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, *2002*, 6–17. https://doi.org/10.1142/9789812799623_0002. (Last accessed on 25/06/2020)

Boutilier, C. (1995). On the Revision of Probabilistic Belief States. *Notre Dame Journal of Formal Logic*, *36*(1), 158–183.

Boutilier, C. (1996). Iterated revision and minimal change of conditional beliefs. *J. Philosophical Logic*, *25*(3), 263–305.

Boutilier, C. (1998). A Unified Model of Qualitative Belief Change: A Dynamical Systems Perspective. *Artif. Intell.*, *98*(1-2), 281–316.

Bradley, D. (2018). Carnap's epistemological critique of metaphysics. *Synthese*, *195*(5), 2247–2265. https://doi.org/10.1007/s11229-017-1335-x. (Last accessed on 25/06/2020)

Buntine, W. (1991). Theory Refinement in Bayesian Networks, In *Proceedings of the 7th Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann Publishers.

Carnap, R. (1950). *The Logical Foundations of Probability*. Chicago, Illinois, U.S.A., The University of Chicago Press.

Cheng, J., Greiner, R., Kelly, J., Bell, D., & Liu, W. (2002). Learning bayesian networks from data: An information-theory based approach. *Artificial Intelligence*, *137*(1), 43–90.

Chhogyal, K. (2015). *Belief change: A probabilistic inquiry* (Doctoral dissertation). Griffith University. Queensland, Australia.

Chickering, D. M. (2003). Optimal Structure Identification with Greedy Search. *J. Mach. Learn. Res.*, 507–554. https://doi.org/10.1162/153244303321897717. (Last accessed on 25/06/2020)

Chickering, D. M., Heckerman, D., & Meek, C. (1997). A bayesian approach to learning bayesian networks with local structure, In *Proceedings of the thirteenth conference on uncertainty in artificial intelligence*, Providence, Rhode Island, Morgan Kaufmann Publishers Inc. http://dl.acm.org/citation.cfm?id=2074226.2074236 (Last accessed on 25/06/2020)

Collins, A., & Michalski, R. (1989). The logic of plausible reasoning: A core theory. *Cognitive Science*, *13*(1), 1–49. https://doi.org/https://doi.org/10.1016/0364-0213(89)90010-4. (Last accessed on 24/06/2020)

Conati, C., Gertner, A. S., VanLehn, K., & Druzdzel, M. J. (1997). On-line student modeling for coached problem solving using bayesian networks (A. Jameson, C. Paris, & C. Tasso, Eds.). In A. Jameson, C. Paris, & C. Tasso (Eds.), *User modeling*, Vienna, Springer Vienna.

Cooper, G. F., & Herskovits, E. (1992). A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, (4), 309–347. https://doi.org/10.1007/BF00994110. (Last accessed on 25/06/2020)

Coutant, A., Leray, P., & Le Capitaine, H. (2014). Learning probabilistic relational models using non-negative matrix factorization. *Proceedings of the 27th International Florida Artificial Intelligence Research Society Conference, FLAIRS 2014*, 490–495.

Craven, M., DiPasquo, D., Freitag, D., & llum, A. M. (1998). Learning to Extract Symbolic Knowledge from the World Wide Web, In *AAAI 1998*. http://www.bibsonomy.org/bibtex/2bfc724ec90047624c8723e8d99929380/bsmyth (Last accessed on 24/06/2020)

da Costa, P. C. G. (2005). *Bayesian semantics for the semantic web* (Doctoral dissertation). George Mason University. Fairfax, Virginia, USA.

da Costa, P. C. G., Laskey, K. B., Laskey, K. J., & Pool, M. (Eds.). (2005, November 9). Uncertainty Reasoning for the Semantic Web, In *ISWC-URSW*. http://dblp.uni-trier.de/db/conf/semweb/ursw2005.html;%20http://www.bibsonomy.org/bibtex/21c94cf97a916985daa06629e59acf26c/dblp (Last accessed on 24/06/2020)

Darwiche, A., & Pearl, J. (1997). On the logic of iterated belief revision. *Artificial Intelligence*, *89*(1), 1–29. https://doi.org/10.1016/S0004-3702(96)00038-0. (Last accessed on 24/06/2020)

Dash, D., & Druzdzel, M. J. (2003). Robust Independence Testing for Constraint-based Learning of Causal Structure, In *Proceedings of the Nineteenth Conference on Uncertainty in Artificial Intelligence*, Acapulco, Mexico, Morgan Kaufmann Publishers Inc. http://dl.acm.org/citation.cfm?id=2100584.2100604 (Last accessed on 24/06/2020)

de Jongh, M., & Druzdzel, M. J. (2009). A comparison of structural distance measures for causal bayesian network models. In M. Klopotek, A. Przepiorkowski, S. T. Wierzchon, & K. Trojanowski (Eds.), *Recent advances in intelligent information systems, challenging problems of science, computer science series* (pp. 443–456). Warsaw, Poland, Academic Publishing House EXIT. http://d-scholarship.pitt.edu/6011/. (Last accessed on 24/06/2020)

Demey, L., Kooi, B., & Sack, J. (2016). Logic and Probability. In E. N. Zalta (Ed.), *The Stanford Encyclopedia of Philosophy* (Winter 2016). Metaphysics Research Lab, Stanford University.

Dempster, A. P. (1968). Upper and lower probabilities generated by a random closed interval. *Ann. Math. Statist.*, *39*(3), 957–966. https://doi.org/10.1214/aoms/1177698328. (Last accessed on 25/06/2020)

Dinh, Q.-T., Vrain, C., & Exbrayat, M. (2012). A Link-Based Method for Propositionalization. (F. Riguzzi & F. Zelezný, Eds.). In F. Riguzzi & F. Zelezný (Eds.), *ILP (Late Breaking Papers)*, CEUR-WS.org. http://dblp.uni-trier.de/db/conf/ilp/ilp2012lbp.html#DinhVE12;%20http://ceur-ws.org/Vol-975/paper-01.pdf;%20http://www.bibsonomy.org/bibtex/23b9c428edaae3af830dee520f4cd6e09/dblp (Last accessed on 24/06/2020)

Doyle, J. (1979). A truth maintenance system. *Artificial Intelligence*, *12*(3), 231–272. https://doi.org/10.1016/0004-3702(79)90008-0. (Last accessed on 24/06/2020)

Dubois, D. (2008). Three Scenarios for the Revision of Epistemic States. *J. Log. and Comput.*, (5), 721–738. https://doi.org/10.1093/logcom/exm092. (Last accessed on 25/06/2020)

Dubois, D., & Prade, H. (1997). Focusing vs. belief revision: A fundamental distinction when dealing with generic knowledge (D. M. Gabbay, R. Kruse, A. Nonnengart, & H. J. Ohlbach, Eds.). In D. M. Gabbay, R. Kruse, A. Nonnengart, & H. J. Ohlbach (Eds.), *Qualitative and Quantitative Practical Reasoning*, Berlin, Heidelberg, Springer Berlin Heidelberg.

Dubois, D., & Prade, H. (2014). Possibility theory. In R. A. Meyers (Ed.), *Encyclopedia of complexity and systems science* (pp. 1–20). New York, NY, Springer New York. https://doi.org/10.1007/978-3-642-27737-5_413-2. (Last accessed on 24/06/2020)

Eggeling, R., Viinikka, J., Vuoksenmaa, A., & Koivisto, M. (2019). On structure priors for learning bayesian networks (K. Chaudhuri & M. Sugiyama, Eds.). In K. Chaudhuri & M. Sugiyama (Eds.), *Proceedings of machine learning research*, PMLR. http://proceedings.mlr.press/v89/eggeling19a.html (Last accessed on 25/06/2020)

Ettouzi, N., Leray, P., & Messaoud, M. B. (2016). An exact approach to learning probabilistic relational model (A. Antonucci, G. Corani, & C. P. Campos, Eds.). In A. Antonucci, G. Corani, & C. P. Campos (Eds.), *Proceedings of the eighth international conference on probabilistic graphical models*.

Fierens, D., Blockeel, H., Bruynooghe, M., & Ramon, J. (2005). Logical Bayesian Networks and Their Relation to Other Probabilistic Logical Models. (S. Kramer & B. Pfahringer, Eds.). In S. Kramer & B. Pfahringer (Eds.), *ILP*, Springer. http://dblp.uni-trier.de/db/conf/ilp/ilp2005.html#FierensBBR05 (Last accessed on 24/06/2020)

Flouris, G., Plexousakis, D., & Antoniou, G. (2006). Evolving Ontology Evolution. *LECTURE NOTES IN COMPUTER SCIENCE*, (3831), 14–29.

França, M. V. M., Zaverucha, G., & d'Avila Garcez, A. S. (2014). Fast relational learning using bottom clause propositionalization with artificial neural networks. *Machine Learning*, *94*(1), 81–104. http://dblp.uni-trier.de/db/journals/ml/ml94.html#FrancaZG14;%20http://dx.doi.org/10.1007/s10994-013-5392-1;

%20http://www.bibsonomy.org/bibtex/2a8aad08efc09976ff4c23c975866cd86/dblp. (Last accessed on 24/06/2020)

Friedman, N., Getoor, L., Koller, D., & Pfeffer, A. (1999). Learning probabilistic relational models, In *Proceedings of the sixteenth international joint conference on artificial intelligence*, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc. http://dl.acm.org/citation.cfm?id=646307.687770 (Last accessed on 25/06/2020)

Friedman, N., & Goldszmidt, M. (1997). Sequential update of bayesian network structure, In *Proceedings of the thirteenth conference on uncertainty in artificial intelligence*, Providence, Rhode Island, Morgan Kaufmann Publishers Inc. http://dl.acm.org/citation.cfm?id=2074226.2074246 (Last accessed on 25/06/2020)

Friedman, N., & Halpern, J. Y. (1994). A Knowledge-Based Framework for Belief Change, Part II: Revision and Update. In J. Doyle, E. Sandewall, & P. Torasso (Eds.), *Principles of Knowledge Representation and Reasoning* (pp. 190–201). Morgan Kaufmann. https://doi.org/10.1016/B978-1-4832-1452-8.50114-7. (Last accessed on 24/06/2020)

Friedman, N., & Halpern, J. Y. (1998). Plausibility Measures and Default Reasoning. http://arxiv.org/abs/cs/9808007v1;%20http://arxiv.org/pdf/cs/9808007v1 (Last accessed on 24/06/2020)

Friedman, N., & Koller, D. (2003). Being bayesian about network structure. a bayesian approach to structure discovery in bayesian networks. *Machine Learning*, *50*(1), 95–125. https://doi.org/10.1023/A:1020249912095. (Last accessed on 25/06/2020)

Gardenfors, P. (1988). *Knowledge in Flux: Modeling the Dynamics of Epistemic States*. MIT Press.

Getoor, L. (2000). Learning Probabilistic Relational Models. *LECTURE NOTES IN COMPUTER SCIENCE*, (1864), 322–323. %7Bhttps://www.ingentaconnect.

com/content/ssam/03029743/2000/00000001/00001864/art00025%7D. (Last accessed on 24/06/2020)

Getoor, L., Friedman, N., Koller, D., & Taskar, B. (2001). Learning Probabilistic Models of Relational Structure. *ICML*.

Getoor, L., Friedman, N., Koller, D., & Taskar, B. (2002). Learning probabilistic models of link structure. *Journal of Machine Learning Research*, *3*, 679–707. https://doi.org/10.1162/jmlr.2003.3.4-5.679. (Last accessed on 25/06/2020)

Goertzel, B. (2014). Artificial general intelligence: Concept, state of the art, and future prospects. *J. Artificial General Intelligence*, *5*, 1–48.

Grdenfors, P., & Makinson, D. (1988). Revisions of Knowledge Systems Using Epistemic Entrenchment, In *Proceedings of the 2Nd Conference on Theoretical Aspects of Reasoning About Knowledge*, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc.

Greaves, H., & Wallace, D. (2006). Justifying conditionalization: Conditionalization maximizes expected epistemic utility. *Mind*, *115*(459), 607–632. http://www.jstor.org/stable/3840583. (Last accessed on 24/06/2020)

Han, S., Zhang, H., Homayouni, R., & Karmaus, W. (2017). An efficient bayesian approach for gaussian bayesian network structure learning. *Communications in Statistics - Simulation and Computation*, *46*(7), 5070–5084. https://doi.org/10.1080/03610918.2016.1143103

Hansson, S. O. (1999). A Survey of Non-Prioritized Belief Revision. *Erkenntnis (1975-)*, *50*(2/3), 413–427. %7Bhttp://www.jstor.org/stable/20012926%7D. (Last accessed on 24/06/2020)

Hansson, S. O. (2017). Logic of belief revision. In E. N. Zalta (Ed.), *The stanford encyclopedia of philosophy* (Winter 2017). Metaphysics Research Lab, Stanford University.

Harper, W. L. (1975). Rational belief change, popper functions and counterfactuals. *Synthese*, *30*(1-2), 221–262.

Heckerman, D. (2008). A tutorial on learning with Bayesian networks. *Innovations in Bayesian Networks*, 33–82.

Heckerman, D., Meek, C., & Koller, D. (2004). *Probabilistic Models for Relational Data* (tech. rep. MSR-TR-2004-30).

Heckerman, D., Geiger, D., & Chickering, D. M. (1995). Learning Bayesian Networks: The Combination of Knowledge and Statistical Data. *Machine Learning*, (3), 197–243. https://doi.org/10.1023/A:1022623210503. (Last accessed on 25/06/2020)

Heckerman, D., Meek, C., & Cooper, G. (2006). One Rogers Street Cambridge, MA 02142-1209, The MIT Press.

Huber, F. (2009). Belief and degrees of belief. Springer. http://philsci-archive.pitt.edu/10851/ (Last accessed on 24/06/2020)

Hunter, A., & Delgrande, J. P. (2005). Iterated Belief Change: A Transition System Approach, In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, Edinburgh, Scotland, Morgan Kaufmann Publishers Inc. %7Bhttp://dl.acm.org/citation.cfm?id=1642293.1642367%7D (Last accessed on 24/06/2020)

Jaeger, M. (1997). *Relational Bayesian Networks.*

Jembere, E., Xulu, S. S., & Adigun, M. O. (2010). Automatic ontology evolution in open and dynamic computing environments (J.-S. Pan, S.-M. Chen, & N. T. Nguyen, Eds.). In J.-S. Pan, S.-M. Chen, & N. T. Nguyen (Eds.), *Computational collective intelligence. technologies and applications*, Berlin, Heidelberg, Springer Berlin Heidelberg.

Jin, Y., & Thielscher, M. (2004). Representing beliefs in the fluent calculus, In *Proceedings of the 16th european conference on artificial intelligence*, Valencia, Spain, IOS Press. http://dl.acm.org/citation.cfm?id=3000001.3000174 (Last accessed on 25/06/2020)

Kadlec, P., Grbić, R., & Gabrys, B. (2011). Review of adaptation mechanisms for data-driven soft sensors. *Computers and Chemical Engineering*, *35*(1), 1–24.

https://doi.org/10.1016/j.compchemeng.2010.07.034. (Last accessed on 24/06/2020)

Katsuno, H., & Mendelzon, A. O. (1991). On the Difference Between Updating a Knowledge Base and Revising It, In *Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning*, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc.

Kemeny, J. G., & Snell, J. L. (1983). New York, Springer-Verlag.

Kersting, K., & Raedt, L. D. (2001). Bayesian Logic Programs. *Arxiv preprint cs.AI/0111058*.

Kersting, K., & Raedt, L. (2008). Basic Principles of Learning Bayesian Logic Programs. In *Probabilistic Inductive Logic Programming*. Springer. https://doi.org/10.1007/978-3-540-78652-8_7. (Last accessed on 24/06/2020)

Keynes, J. M. (1921). *A treatise on probability*. London, Macmillan.

Kolmogorov, A. N. (1956). One Rogers Street Cambridge, MA 02142-1209, Chelsea Publishing Company.

Korb, K. B., & Nicholson, A. E. (2004). *Bayesian artificial intelligence* (Vol. 1). cRc Press.

Lam, W., & Bacchus, F. (1994). Using new data to refine a bayesian network, In *Proceedings of the tenth international conference on uncertainty in artificial intelligence*, Seattle, WA, Morgan Kaufmann Publishers Inc. http://dl.acm.org/citation.cfm?id=2074394.2074443 (Last accessed on 25/06/2020)

Lang, J. (2007). Belief update revisited, In *In Proceedings of the International Joint Conference on Artifi- cial Intelligence*.

Lange, T., Roth, V., Braun, M. L., & Buhmann, J. M. (2004). Stability-based validation of clustering solutions. *Neural Computation*, *16*(6), 1299–1323. https://doi.org/10.1162/089976604773717621. (Last accessed on 25/06/2020)

Laskey, K. B. (2008). MEBN: A Language for First-Order Bayesian Knowledge Bases. *Artif. Intell.*, *172*(2-3), 140–178.

Lauritzen, S. L., & Spiegelhalter, D. J. (1990). Readings in uncertain reasoning. In G. Shafer & J. Pearl (Eds.). San Francisco, CA, USA, Morgan Kaufmann

Publishers Inc. http://dl.acm.org/citation.cfm?id=84628.85343. (Last accessed on 24/06/2020)

Lee, S. M., & Kim, S. B. (2019). Identification of i-equivalent subnetworks in bayesian networks to incorporate experts' knowledge. *Expert Systems*, *36*(1), https://onlinelibrary.wile e12346. https://doi.org/10.1111/exsy.12346. (Last accessed on 25/06/2020)

Levi, I. (1977). Subjunctives, dispositions and chances. *Synthese*, *34*(4), 423–455. http://www.jstor.org/stable/20115172. (Last accessed on 25/06/2020)

Lewis, D. (1976). Probabilities of Conditionals and Conditional Probabilities. *The Philosophical Review*, *85*(3), 297–315. %7Bhttp://www.jstor.org/stable/2184045%7D. (Last accessed on 24/06/2020)

Lindström, S., & Rabinowicz, W. (1989). On probabilistic representation of non-probabilistic belief revision. *J. Philosophical Logic*, *18*(1), 69–101.

Liu, W., Yue, K., Yue, M., Yin, Z., & Zhang, B. (2018). A bayesian network-based approach for incremental learning of uncertain knowledge. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, *26*, 87–108. https://doi.org/10.1142/S021848851850006X. (Last accessed on 25/06/2020)

Lukasiewicz, T. (2002). Probabilistic Default Reasoning with Conditional Constraints. *Ann. Math. Artif. Intell.*, *34*(1-3), 35–88. http://dblp.uni-trier.de/db/journals/amai/amai34.html#Lukasiewicz02;%20http://dx.doi.org/10.1023/A:1014445017537. (Last accessed on 24/06/2020)

Madigan, D., & Raftery, A. E. (1994). Model selection and accounting for model uncertainty in graphical models using occam's window. *Journal of the American Statistical Association*, *89*(428), 1535–1546.

Makinson, D. (1987). On the status of the postulate of recovery in the logic of theory change. *Journal of Philosophical Logic*, *16*(4), 383–394. http://www.jstor.org/stable/30226379

Maurice, P., & David, R. (2005). *Inverse Resolution as Belief Change* (Leslie Pack Kaelbling and Alessandro Saffiotti, Ed.), Professional Book Center, *IJCAI-05, Proceedings of the Nineteenth International Joint Conference on Artificial*

Intelligence, Edinburgh, Scotland, UK, July 30 - August 5, 2005, 534–539. %7Bhttp://ijcai.org/proceedings/2005%7D (Last accessed on 24/06/2020)

McCallum, A. (2017). *Cora Dataset*. Texas Data Repository Dataverse. http://dx.doi.org/10.18738/T8/HUIG48 (Last accessed on 24/06/2020)

Meyer, T., Ghose, A., & Chopra, S. (2001). Non-prioritised Ranked Belief Change, 151–162. http://dl.acm.org/citation.cfm?id=1028128.1028145. (Last accessed on 24/06/2020)

Milch, B., Marthi, B., Russell, S., Sontag, D., Ong, D. L., & Kolobov, A. (2005). BLOG: Probabilistic Models with Unknown Objects, In *Proc. 19th International Joint Conf. Artificial Intelligence (IJCAI-05)*, Edinburgh.

Moreno-Torres, J. G., Raeder, T., Alaiz-Rodrıguez, R., Chawla, N. V., & Herrera, F. (2012). A unifying view on dataset shift in classification. *Pattern Recognition*, *45*(1), 521–530. https://doi.org/10.1016/j.patcog.2011.06.019. (Last accessed on 24/06/2020)

Možina, M., Guid, M., Krivec, J., Sadikov, A., & Bratko, I. (2008). Fighting knowledge acquisition bottleneck with argument based machine learning, In *Proceedings of the 18th European Conference on Artificial Intelligence (ECAI'08)*.

Muggleton, S. (1991). Inductive Logic Programming. *New Generation Computing*, *8*(4), 295–318.

Muggleton, S. (1996). Stochastic Logic Programs, In *Advances in Inductive Logic Programming*.

Natarajan, S., Wong, W., & Tadepalli, P. (2006). Structure refinement in First Order Conditional Influence Language, In *Proceedings of the ICML workshop on Open Problems in Statistical Relational Learning*.

Nau, R. F. (2001). De finetti was right: Probability does not exist. *Theory and Decision*, *51*(2), 89–124. https://doi.org/10.1023/A:1015525808214. (Last accessed on 24/06/2020)

Neapolitan, R. E. (1990). *Probabilistic Reasoning in Expert Systems: Theory and Algorithms*. New York, NY, USA, John Wiley & Sons, Inc.

Niinimaki, T., & Parviainen, P. (2012). Local structure discovery in bayesian networks, In *Proceedings of the twenty-eighth conference on uncertainty in artificial intelligence*, Catalina Island, CA, AUAI Press. http://dl.acm.org/citation.cfm?id=3020652.3020719 (Last accessed on 25/06/2020)

Onisko, A. (2003). *Probabilistic causal models in medicine: Application to diagnosis of liver disorders* (Doctoral dissertation). Polish Academy of Science. Warsaw, Poland.

Ourston, D., & Mooney, R. J. (1994). Theory refinement combining analytical and empirical methods. *Artificial Intelligence*, *66*(2), 273–309. https://doi.org/https://doi.org/10.1016/0004-3702(94)90028-0. (Last accessed on 24/06/2020)

Palmisano, I., Tamma, V. A. M., Iannone, L., Payne, T. R., & Doran, P. (2008). Dynamic Ontology Evolution in Open Environments. (C. Bizer & A. Joshi, Eds.). In C. Bizer & A. Joshi (Eds.), *International Semantic Web Conference (Posters and Demos)*, CEUR-WS.org. http://dblp.uni-trier.de/db/conf/semweb/iswc2008p.html#PalmisanoTIPD08;%20http://ceur-ws.org/Vol-401/iswc2008pd_submission_41.pdf (Last accessed on 24/06/2020)

Park, C. Y., & Laskey, K. B. (2018). Human-aided multi-entity bayesian networks learning from relational data. *CoRR*, *abs/1806.02421* arXiv 1806.02421. http://arxiv.org/abs/1806.02421. (Last accessed on 24/06/2020)

Park, C. Y., Laskey, K. B., Costa, P., & Matsumoto, S. (2013). Multi-entity bayesian networks learning in predictive situation awareness., In *In: International Command and Control Research and Technology Symposium (ICCRTS)*.

Park, C. Y., Laskey, K. B., Costa, P. C. G., & Matsumoto, S. (2014). Predictive situation awareness reference model using Multi-Entity Bayesian Networks., In *FUSION*, IEEE.

Park, C. Y., Laskey, K. B., Costa, P. C. G., & Matsumoto, S. (2016). A process for human-aided Multi-Entity Bayesian Networks learning in Predictive Situation Awareness., In *FUSION*, IEEE.

Park, C. Y., Laskey, K. B., da Costa, P. C. G., & Matsumoto, S. (2013). Multi-Entity Bayesian Networks learning for hybrid variables in situation awareness., In *FUSION*, IEEE.

Pearl, J. (2009). *Probabilistic reasoning in intelligent systems : networks of plausible inference*. San Francisco, Calif., Morgan Kaufmann. Example for Explaning away.

Peter Spirtes, C. G., & Scheines, R. (2000). One Rogers Street Cambridge, MA 02142-1209, The MIT Press.

Popescul, A., & Ungar, L. H. (2003). Statistical relational learning for link prediction.

Rens, G., & Meyer, T. A. (2015). A new approach to probabilistic belief change, In *Flairs conference*.

Richardson, M., & Domingos, P. (2006). Markov Logic Networks. *Machine Learning*, *62*, 107–136.

Roelofs, H. D. (1935). A Note on Probability and Ontology. *The Philosophical Review*, (5), 484–487.

Rosinger, E. E. (2010). Quantum foundations : Is probability ontological ? arXiv: physics/0703019v2. (Last accessed on 24/06/2020)

Rott, H. (2000). Two dogmas of belief revision. *Journal of Philosophy*, *97*(9), 503–522.

Shafer, G. (1976). *A Mathematical Theory of Evidence*. Princeton University Press.

Shani, G. (2007). *Learning and solving partially observable markov decision processes* (Doctoral dissertation). Ben-Gurion University of the Negev. Beersheba, Israel.

Shapiro, S., & Pagnucco, M. (2004). Iterated Belief Change and Exogenous Actions in the Situation Calculus, In *Proceedings of the 16th European Conference on Artificial Intelligence*, Amsterdam, The Netherlands, The Netherlands.

Spohn, W. (1988). Ordinal Conditional Functions: A Dynamic Theory of Epistemic States. In *Causation in Decision, Belief Change, and Statistics*. Springer. https://doi.org/10.1007/978-94-009-2865-7_6. (Last accessed on 24/06/2020)

Spohn, W. (2012). One Rogers Street Cambridge, MA 02142-1209, Oxford University Press.

Stojanovic, L., Maedche, A., Stojanovic, N., & Studer, R. (2003). Ontology evolution as reconfiguration-design problem solving, In *Proceedings of the 2nd international conference on knowledge capture*, Sanibel Island, FL, USA, ACM. https://doi.org/10.1145/945645.945669 (Last accessed on 24/06/2020)

Talvitie, T., Eggeling, R., & Koivisto, M. (2018). Finding optimal bayesian networks with local structure (V. Kratochvíl & M. Studený, Eds.). In V. Kratochvíl & M. Studený (Eds.), *Proceedings of the ninth international conference on probabilistic graphical models*, Prague, Czech Republic, PMLR. http://proceedings.mlr.press/v72/talvitie18a.html (Last accessed on 25/06/2020)

Taskar, B., Abbeel, P., Wong, M.-F., & Koller, D. (2007). Relational Markov Networks. In *Introduction to Statistical Relational Learning*.

Tsamardinos, I., Brown, L. E., & Aliferis, C. F. (2006). The max-min hill-climbing bayesian network structure learning algorithm. *Machine Learning*, *65*(1), 31–78. https://doi.org/10.1007/s10994-006-6889-7. (Last accessed on 24/06/2020)

Verma, T., & Pearl, J. (1991). Equivalence and synthesis of causal models, In *Proceedings of the sixth annual conference on uncertainty in artificial intelligence*, New York, NY, USA, Elsevier Science Inc. http://dl.acm.org/citation.cfm?id=647233.719736 (Last accessed on 25/06/2020)

Williams, J. D., & Young, S. (2007). Partially observable markov decision processes for spoken dialog systems. *Computer Speech & Language*, *21*(2), 393–422.

Williams, P. M. (1980). Bayesian conditionalisation and the principle of minimum information. *British Journal for the Philosophy of Science*, *31*(2), 131–144.

Yu, H. (2019). Incremental learning of bayesian networks from concept-drift data, In *2019 ieee 4th international conference on cloud computing and big data analysis (icccbda)*. https://doi.org/10.1109/ICCCBDA.2019.8725689 (Last accessed on 25/06/2020)

Yue, K., Fang, Q., Wang, X., Li, J., & Liu, W. (2015). A parallel and incremental approach for data-intensive learning of bayesian networks. *IEEE Transactions on Cybernetics*, *45*(12), 2890–2904. https://doi.org/10.1109/TCYB.2015.2388791. (Last accessed on 25/06/2020)

Zadeh, L. A. (1977). Possibility theory vs. probability theory in decision analysis, In *1977 ieee conference on decision and control including the 16th symposium on adaptive processes and a special symposium on fuzzy set theory and applications*. https://doi.org/10.1109/CDC.1977.271764 (Last accessed on 24/06/2020)

Zhao, Y., Chen, Y., Tu, K., & Tian, J. (2015). Curriculum learning of bayesian network structures, In *Acml*.

Žliobaitė, I., Pechenizkiy, M., & Gama, J. (2016). An Overview of Concept Drift Applications. In *Big Data Analysis: New Algorithms for a New Society*. Springer. https://doi.org/10.1007/978-3-319-26989-4_4. (Last accessed on 24/06/2020)

APPENDICES

A. XML Schemas and Files

A.1 Epistemic State Schema

```
<?xml version="1.0" encoding="UTF-8" ?>
<!--
This schema defines the data model for the holding the
Epistemic State for the Unified Belief Change Model
-->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
            xmlns:tns="http://www.uzulu.ac.za/2013/EpistemicState"
            elementFormDefault="qualified"
            targetNamespace="http://www.uzulu.ac.za/2013/EpistemicState"
            xmlns="http://www.uzulu.ac.za/2013/EpistemicState">
    <xs:element name="NetworkStructure">
        <xs:complexType>
            <xs:sequence>
                <xs:element name = "variables" minOccurs="0" maxOccurs="unbounded">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name = "lags" minOccurs="0" maxOccurs="unbounded">
                                <xs:complexType>
                                    <xs:sequence>
                                        <xs:element name = "edges" minOccurs="0" maxOccurs="unbounded">
                                            <xs:complexType>
                                                <xs:attribute name = "parentVarIndex" type="xs:int"/>

                                            </xs:complexType>
                                        </xs:element>
                                    </xs:sequence>
                                    <xs:attribute name = "lagIndex" type = "xs:int"/>
                                    <xs:attribute name = "parentCount" type = "xs:int"/>
                                </xs:complexType>
                            </xs:element>
                        </xs:sequence>
                        <xs:attribute name = "varIndex" type = "xs:int"/>
                    </xs:complexType>
```

```
                </xs:element>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="epistemicStateSchema">
        <xs:complexType>
            <xs:sequence>
                 <xs:element name ="BeliefSet">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element ref = "NetworkStructure"/>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
                <xs:element name ="BestNetworkStructure">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element ref = "NetworkStructure"/>
                        </xs:sequence>
                        <xs:attribute name="NetworkScore" type = "xs:double"/>
                    </xs:complexType>
                </xs:element>
                <xs:element name ="BeliefState">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name ="structure"  minOccurs ="0"
                                maxOccurs = "unbounded">
                                <xs:complexType>
                                    <xs:sequence>
                                        <xs:element ref = "NetworkStructure"/>
                                    </xs:sequence>
                                    <xs:attribute name="probability" type = "
                                        xs:double"/>
                                </xs:complexType>
                            </xs:element>
                        </xs:sequence>
                     </xs:complexType>
                </xs:element>
                <xs:element name="EdgesLikelihoods">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name = "wVariables" minOccurs="0"
                                maxOccurs="unbounded">
                                <xs:complexType>
```

```
                        <xs:sequence>
                            <xs:element name = "WLags" minOccurs="0" maxOccurs="unbounded">
                                <xs:complexType>
                                    <xs:sequence>
                                        <xs:element name = "wEdges" minOccurs="0" maxOccurs="unbounded">
                                            <xs:complexType>
                                                <xs:attribute name = "WParentVarIndex" type="xs:int" />
                                                <xs:attribute name = "edgeLikelihood" type ="xs:double" />
                                            </xs:complexType>
                                        </xs:element>
                                    </xs:sequence>
                                    <xs:attribute name = "wLagIndex" type = "xs:int"/>
                                    <xs:attribute name = "wParentCount" type = "xs:int" />
                                </xs:complexType>
                            </xs:element>
                        </xs:sequence>
                        <xs:attribute name = "wVarIndex" type = "xs:int" />
                    </xs:complexType>
                </xs:element>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
        </xs:sequence>
        <xs:attribute name = "varCount" type= "xs:int"/>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

A.2 Epistemic State Except for the ASIA Network

```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<epistemicStateSchema xmlns="http://www.uzulu.ac.za/2013/EpistemicState"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="
    http://www.uzulu.ac.za/2013/EpistemicState /home/banjo/Asia/xmlDir/
    epistemicStateSchema.xsd">
    <BeliefSet>
        <NetworkStructure>
            <variables varIndex="0">
                <lags lagIndex="0"/>
            </variables>
            <variables varIndex="1">
                <lags lagIndex="0"/>
            </variables>
            <variables varIndex="2">
                <lags lagIndex="0"/>
            </variables>
            <variables varIndex="3">
                <lags lagIndex="0"/>
            </variables>
            <variables varIndex="4">
                <lags lagIndex="0">
                    <edges parentVarIndex="2"/>
                </lags>
            </variables>
            <variables varIndex="5">
                <lags lagIndex="0">
                    <edges parentVarIndex="1"/>
                    <edges parentVarIndex="3"/>
                </lags>
            </variables>
            <variables varIndex="6">
                <lags lagIndex="0">
                    <edges parentVarIndex="5"/>
                </lags>
            </variables>
            <variables varIndex="7">
                <lags lagIndex="0">
                    <edges parentVarIndex="4"/>
                    <edges parentVarIndex="5"/>
                </lags>
            </variables>
        </NetworkStructure>
```

```xml
</BeliefSet>
<BestNetworkStructure NetworkScore="-6412.1211051169375">
    <NetworkStructure>
        <variables varIndex="0">
            <lags lagIndex="0"/>
        </variables>
        <variables varIndex="1">
            <lags lagIndex="0"/>
        </variables>
        <variables varIndex="2">
            <lags lagIndex="0"/>
        </variables>
        <variables varIndex="3">
            <lags lagIndex="0">
                <edges parentVarIndex="2"/>
            </lags>
        </variables>
        <variables varIndex="4">
            <lags lagIndex="0">
                <edges parentVarIndex="2"/>
            </lags>
        </variables>
        <variables varIndex="5">
            <lags lagIndex="0">
                <edges parentVarIndex="1"/>
                <edges parentVarIndex="3"/>
            </lags>
        </variables>
        <variables varIndex="6">
            <lags lagIndex="0">
                <edges parentVarIndex="5"/>
            </lags>
        </variables>
        <variables varIndex="7">
            <lags lagIndex="0">
                <edges parentVarIndex="4"/>
                <edges parentVarIndex="5"/>
            </lags>
        </variables>
    </NetworkStructure>
</BestNetworkStructure>
<BeliefState>
    <structure probability="0.03042598122991598">
        <NetworkStructure>
```

```
                <variables varIndex="0">
                    <lags lagIndex="0"/>
                </variables>
                <variables varIndex="1">
                    <lags lagIndex="0"/>
                </variables>
                <variables varIndex="2">
                    <lags lagIndex="0"/>
                </variables>
                <variables varIndex="3">
                    <lags lagIndex="0">
                        <edges parentVarIndex="2"/>
                    </lags>
                </variables>
                <variables varIndex="4">
                    <lags lagIndex="0">
                        <edges parentVarIndex="2"/>
                    </lags>
                </variables>
                <variables varIndex="5">
                    <lags lagIndex="0">
                        <edges parentVarIndex="1"/>
                        <edges parentVarIndex="3"/>
                    </lags>
                </variables>
                <variables varIndex="6">
                    <lags lagIndex="0">
                        <edges parentVarIndex="5"/>
                    </lags>
                </variables>
                <variables varIndex="7">
                    <lags lagIndex="0">
                        <edges parentVarIndex="4"/>
                        <edges parentVarIndex="5"/>
                    </lags>
                </variables>
            </NetworkStructure>
        </structure>
        <structure probability="0.030389117610693774">
            <NetworkStructure>
                <variables varIndex="0">
                    <lags lagIndex="0"/>
                </variables>
                <variables varIndex="1">
```

```
              <lags lagIndex="0"/>
          </variables>
          <variables varIndex="2">
              <lags lagIndex="0"/>
          </variables>
          <variables varIndex="3">
              <lags lagIndex="0">
                  <edges parentVarIndex="2"/>
              </lags>
          </variables>
          <variables varIndex="4">
              <lags lagIndex="0">
                  <edges parentVarIndex="2"/>
              </lags>
          </variables>
          <variables varIndex="5">
              <lags lagIndex="0">
                  <edges parentVarIndex="0"/>
                  <edges parentVarIndex="1"/>
                  <edges parentVarIndex="3"/>
              </lags>
          </variables>
          <variables varIndex="6">
              <lags lagIndex="0">
                  <edges parentVarIndex="5"/>
              </lags>
          </variables>
          <variables varIndex="7">
              <lags lagIndex="0">
                  <edges parentVarIndex="4"/>
                  <edges parentVarIndex="5"/>
              </lags>
          </variables>
      </NetworkStructure>
  </structure>

     ...
     ...
     ...

    <structure probability="0.03025682906252734">
      <NetworkStructure>
          <variables varIndex="0">
              <lags lagIndex="0"/>
```

```
            </variables>
            <variables varIndex="1">
                <lags lagIndex="0">
                    <edges parentVarIndex="0"/>
                </lags>
            </variables>
            <variables varIndex="2">
                <lags lagIndex="0"/>
            </variables>
            <variables varIndex="3">
                <lags lagIndex="0">
                    <edges parentVarIndex="2"/>
                </lags>
            </variables>
            <variables varIndex="4">
                <lags lagIndex="0">
                    <edges parentVarIndex="2"/>
                </lags>
            </variables>
            <variables varIndex="5">
                <lags lagIndex="0">
                    <edges parentVarIndex="1"/>
                    <edges parentVarIndex="3"/>
                </lags>
            </variables>
            <variables varIndex="6">
                <lags lagIndex="0">
                    <edges parentVarIndex="0"/>
                    <edges parentVarIndex="5"/>
                </lags>
            </variables>
            <variables varIndex="7">
                <lags lagIndex="0">
                    <edges parentVarIndex="4"/>
                    <edges parentVarIndex="5"/>
                </lags>
            </variables>
        </NetworkStructure>
    </structure>
</BeliefState>
<EdgesLikelihoods>
    <wVariables wVarIndex="0">
        <WLags wLagIndex="0">
            <wEdges WParentVarIndex="0" edgeLikelihood="0.0"/>
```

```xml
            <wEdges WParentVarIndex="1" edgeLikelihood="0.03030158589207877"/>
            <wEdges WParentVarIndex="2" edgeLikelihood="0.0"/>
            <wEdges WParentVarIndex="3" edgeLikelihood="0.0"/>
            <wEdges WParentVarIndex="4" edgeLikelihood="0.0"/>
            <wEdges WParentVarIndex="5" edgeLikelihood="0.030292563729169768"/>
            <wEdges WParentVarIndex="6" edgeLikelihood="0.030296752196502998"/>
            <wEdges WParentVarIndex="7" edgeLikelihood="0.06060030004553526"/>
        </WLags>
    </wVariables>
    <wVariables wVarIndex="1">
        <WLags wLagIndex="0">
            <wEdges WParentVarIndex="0" edgeLikelihood="0.1817424839006842"/>
            <wEdges WParentVarIndex="1" edgeLikelihood="0.0"/>
            <wEdges WParentVarIndex="2" edgeLikelihood="0.03030158589207877"/>
            <wEdges WParentVarIndex="3" edgeLikelihood="0.0"/>
            <wEdges WParentVarIndex="4" edgeLikelihood="0.0"/>
            <wEdges WParentVarIndex="5" edgeLikelihood="0.0"/>
            <wEdges WParentVarIndex="6" edgeLikelihood="0.0"/>
            <wEdges WParentVarIndex="7" edgeLikelihood="0.0"/>
        </WLags>
    </wVariables>
    <wVariables wVarIndex="2">
        <WLags wLagIndex="0">
            <wEdges WParentVarIndex="0" edgeLikelihood="0.0"/>
            <wEdges WParentVarIndex="1" edgeLikelihood="0.21205663846048245"/>
            <wEdges WParentVarIndex="2" edgeLikelihood="0.0"/>
            <wEdges WParentVarIndex="3" edgeLikelihood="0.18167833270904585"/>
            <wEdges WParentVarIndex="4" edgeLikelihood="0.03025696376425163"/>
            <wEdges WParentVarIndex="5" edgeLikelihood="0.060559336582042184"/>
            <wEdges WParentVarIndex="6" edgeLikelihood="0.0"/>
            <wEdges WParentVarIndex="7" edgeLikelihood="0.0"/>
        </WLags>
    </wVariables>
    <wVariables wVarIndex="3">
        <WLags wLagIndex="0">
```

```
          <wEdges WParentVarIndex="0" edgeLikelihood="0.06055838212248253"/>
          <wEdges WParentVarIndex="1" edgeLikelihood="0.03029088758862727"/>
          <wEdges WParentVarIndex="2" edgeLikelihood="0.818321667290954"/>
          <wEdges WParentVarIndex="3" edgeLikelihood="0.0"/>
          <wEdges WParentVarIndex="4" edgeLikelihood="0.0"/>
          <wEdges WParentVarIndex="5" edgeLikelihood="0.0"/>
          <wEdges WParentVarIndex="6" edgeLikelihood="0.0"/>
          <wEdges WParentVarIndex="7" edgeLikelihood="0.0"/>
      </WLags>
  </wVariables>
  <wVariables wVarIndex="4">
      <WLags wLagIndex="0">
          <wEdges WParentVarIndex="0" edgeLikelihood="0.03025696376425163"/>
          <wEdges WParentVarIndex="1" edgeLikelihood="0.0"/>
          <wEdges WParentVarIndex="2" edgeLikelihood="0.9697430362357483"/>
          <wEdges WParentVarIndex="3" edgeLikelihood="0.0"/>
          <wEdges WParentVarIndex="4" edgeLikelihood="0.0"/>
          <wEdges WParentVarIndex="5" edgeLikelihood="0.0"/>
          <wEdges WParentVarIndex="6" edgeLikelihood="0.0"/>
          <wEdges WParentVarIndex="7" edgeLikelihood="0.0"/>
      </WLags>
  </wVariables>
  <wVariables wVarIndex="5">
      <WLags wLagIndex="0">
          <wEdges WParentVarIndex="0" edgeLikelihood="0.27271436698625406"/>
          <wEdges WParentVarIndex="1" edgeLikelihood="1.0"/>
          <wEdges WParentVarIndex="2" edgeLikelihood="0.03030215397459654"/>
          <wEdges WParentVarIndex="3" edgeLikelihood="1.0"/>
          <wEdges WParentVarIndex="4" edgeLikelihood="0.030296017273059552"/>
          <wEdges WParentVarIndex="5" edgeLikelihood="0.0"/>
          <wEdges WParentVarIndex="6" edgeLikelihood="0.0"/>
          <wEdges WParentVarIndex="7" edgeLikelihood="0.0"/>
      </WLags>
  </wVariables>
  <wVariables wVarIndex="6">
      <WLags wLagIndex="0">
          <wEdges WParentVarIndex="0" edgeLikelihood="0.12113181404461136"/>
```

```
                <wEdges WParentVarIndex="1" edgeLikelihood="0.15147144731735712"/>
                <wEdges WParentVarIndex="2" edgeLikelihood="0.0"/>
                <wEdges WParentVarIndex="3" edgeLikelihood="0.1514685186788411"/>
                <wEdges WParentVarIndex="4" edgeLikelihood="0.0"/>
                <wEdges WParentVarIndex="5" edgeLikelihood="0.9697120006842175"/>
                <wEdges WParentVarIndex="6" edgeLikelihood="0.0"/>
                <wEdges WParentVarIndex="7" edgeLikelihood="0.0"/>
            </WLags>
        </wVariables>
        <wVariables wVarIndex="7">
            <WLags wLagIndex="0">
                <wEdges WParentVarIndex="0" edgeLikelihood="0.0"/>
                <wEdges WParentVarIndex="1" edgeLikelihood="0.0"/>
                <wEdges WParentVarIndex="2" edgeLikelihood="0.0"/>
                <wEdges WParentVarIndex="3" edgeLikelihood="0.0"/>
                <wEdges WParentVarIndex="4" edgeLikelihood="1.0"/>
                <wEdges WParentVarIndex="5" edgeLikelihood="1.0"/>
                <wEdges WParentVarIndex="6" edgeLikelihood="0.0"/>
                <wEdges WParentVarIndex="7" edgeLikelihood="0.0"/>
            </WLags>
        </wVariables>
    </EdgesLikelihoods>
</epistemicStateSchema>
```

B. An Example R Scripts for simulating data in Propositional Bayesian Networks

```
simulateData = function(){
    library(bnlearn)
    asia<- read.bif("/home/edgar/Banjo/Asia/30runs/asia.bif")
    aData <-rbn(asia, 1000)
    res = empty.graph(names(aData))
    modelstring(res) = paste("[asia]",
        "[smoke]",
        "[tub|asia]",
        "[lung|smoke]",
        "[bronc|smoke]",
        "[either|lung:tub]",
        "[xray|either]",
        "[dysp|bronc:either]",
        sep = "")
    sim <-rbn(res, 3000, aData)
    data <-as.matrix(sapply(sim, as.numeric))
    write.table(na.omit(data), file ="/home/edgar/Banjo/Asia/30runs/saveddf.txt",
         sep ="\t", row.names = F, col.names = F)
}
```