University of Cape Town

-

An Overview of Defeasible Implication within Bayesian Networks

Elijah Roussos
RSSELI007

ABSTRACT

In this paper we examine the potential intersection of Bayesian Networks and non-monotonic Logic, with the goal of supplementing a Bayesian Reasoner with a non-monotonic knowledge base in order to reduce Bayesian Network complexity. Specifically, we focus on the properties, limitations and reasoning methodologies of both areas, and the theory of how a combined reasoner may function. We review 3 fundamental areas of preceding research: Propositional Logic and Classical Reasoning, Defeasiblility and Non-monotonic Reasoning in Propositional Logic, Bayesian Networks and Bayesian Reasoning, as well as investigate the potential impact of logical implication in Bayesian Reasoning. We recognise from the literature the benefit and the efficiency of propositional and non-monotonic logic systems in being able to depict casual relationships between objects. In particular, we propose propositional implication may have obvious synergy with the independent variables of a Bayesian Network. We deduce that if a logical consequence is specified between two variables of a Bayesian Network, those variables may collapse into a single, composite variable in the network, simplifying the network. Furthermore, while we identify the benefit of adding defeasibility to this construct in being able to specify contextual relationships in a Bayesian Network as opposed to strict relationships, it is not as obvious what the consequence would be on the network variables. Overall, we conclude that further research is required into the role of logic and defeasibility in Bayesian Network reasoners, as the theory behind such a combined system is largely underdeveloped and the benefits of which are unknown.

Keywords

Propositional Logic; Knowledge Representation;
Defeasible Reasoning; Non-Monotonic Logic;
Bayesian Networks; Artificial Intelligence;

1. INTRODUCTION

Artificial intelligence (AI) has in recent years grown significantly both in depth and breadth, due to the growing need for systems that are able solve increasingly more complex tasks. This boom has accelerated development in a variety of sub-areas of the field, stemming from computer vision to machine learning. One such field is known as knowledge representation and reasoning, a form of logic based AI where information about a system, known as a knowledge base, is used solve complex decision problems [1].

Traditionally, such systems were based off propositional logic, a mathematical formalism wherein statements, or propositions, about actors in a system are used to draw conclusions. However, propositional logic reasoners are limiting in a number of ways. In particular, these systems are poor at dealing with exceptionality within the knowledge base of the problem domain. That is, given a propositional implication such as "a implies b", in order to infer a conclusion from the associated knowledge base, the relationship between the two atoms in the statement should be definitively casual. This fact limits the reasoning capability of classical knowledge bases, as statements such as "a **may** imply b" are simply not defined. This shortfall is addressed directly by introducing the notion of defeasiblity, that is statements that present a typical rather than direct casual relationship. In a defeasible reasoner, typicality allows for reasoning about logical statements that would otherwise be in conflict with each other. As such, exceptional statements such as "a typically implies b" are allowed [2]. Nonetheless, propositional logic is binary, and is unable to reason with probabilistic systems.

Another branch of reasoning, known as Bayesian reasoning, uses a what is known as Bayesian Network to calculate the likelihood of a certain event occurring given the relationships between atoms within a system and their associated probabilities [3]. Bayesian Networks provide a powerful mechanism for reasoning with probabilities, as a variety of efficient algorithms have been developed to calculate the probability of an event occurring in the system agnostic of the underlying probability distribution. These algorithms rely on a key property of a Bayesian Network: atoms in a network become independent of its non-descendents once the atoms which are causal to it, or parents, become known. While there exists a multitude of research in the literature for defeasible reasoning and Bayesian Networks, few papers have explored the idea of combining the techniques to create a single knowledge base that can reason with both defeasible statements and probabilistic statements.

This literature review aims to investigate the combination of these two approaches, predominately by supplementing a Bayesian Network system with either a propositional or non-monotonic knowledge base, with the goal of reducing the complexity of a Bayesian Network and ultimately speeding up the computation time of Bayesian reasoning algorithms. In reviewing these areas, we seek to determine the viability and practicality of such a reasoner by understanding the limitations, advantages and uses of each reasoning methodology. This literature review will cover four main topics of interest to the area. These include an overview of classical logic, defeasibility in logic and defeasible reasoning, the theory surrounding Bayesian Networks and reasoning in a Bayesian Network and the consequence of logical implication within a Bayesian Network. Finally, we will discuss the idea of extending a Bayesian Network with either a classical or defeasible knowledge base, what the interaction between the systems may entail.

2. CLASSICAL REASONING

There exist various mechanishms that allow us to make inferences about the systems that govern the world around us. One such mechanism, logic, formalises this process, providing a mathematical basis for drawing logical conclusions from a collection of premises. Indeed, logic does not necessarily comply with one particular representation, and as such there exist a variety of "logics", each providing a certain level of expressivity and usefulness in particular domains. However, expressivity is not without cost, as more expressive logics are typically more complex to evaluate algorithmically [4].

The notion of reasoning in logic means to entail certain conclusions from what is known as a knowledge base, i.e. a collection of statements in a logic language describing the interactions and relationships between actors in a system. Thus, given a statement s and a knowledge base K, a reasoner will determine whether s is consistent with K [7]. If s can indeed be concluded from K, we denote entailment, otherwise known as logical consequence, with the symbol $\models$ and write:

$$K \models s\ .$$

2.1 Propositional Logic

When people use logic to make decisions in the real world, the conclusions drawn are typically based on facts about the environment in question. In particular, people determine whether or not a particular "thing" is true or false in a certain context, draw relationships between these different "things" and use that information to make inferences. For example, one might infer after recently making a cup of tea that if you were to drink it, you would burn yourself as the tea is hot. In this case, a logical connection has been made between tea and being burnt, based on the proposition that we know hot items burn things and the tea is hot.

Formally, propositional logic is a mathematical language, used to reason about claims that cannot be further decomposed and have a value of either true or false. It provides the basis for all other logics [5]. To construct a propositional sentence we form compositions of boolean variables, otherwise known as atoms (or "things" in our analogy). While an atom is itself a propositional sentence (specifically an atomic sentence), more complicated sentences may be formed altering atoms or stating specific relationships relationships between atoms. A sentence is said to be *declarative* should it correspond to a specific truth value. A collection of a number of these declarative sentences corresponds to a propositional knowledge base.

2.1.1 Boolean Variables

In order to demonstrate the language of propositional logic, suppose the previous tea example. In this example, our atoms consist of tea, hot and burn, which in classical logic are boolean variables which could be represented by the letters T, H and B. Therefore where the object in question is X, $T(X)$ implies the object *is* tea, $H(X)$ implies the object *is* hot and $B(X)$ implies the object *is* burning [11]. For simplicity, the (X) in the atoms can be omitted, such that $H(X) = H$. As mentioned above however, there is little that can be concluded from the atoms themselves without stating the relationships that exist between them. In order to do so, propositional logic makes use of logical connectives, of which 6 exist.

To demonstrate how these connectives work, they will each be depicted in what is known as a *truth table*. Truth tables provide a useful insight into how variables in such a system interact by presenting what each possible sentence, or world, of the system evaluates to. If there are n atoms in the system, there will be 2^n possible worlds with valid evaluations. Using truth tables, the validity of logical sentences describing the system can be checked. [6]

2.1.2 Logical Connectives

The content of the following section(2.1.2) cites Darwiche, A et al [6] (Aside from the xor operation).

Consider the variable $T(X)$. We know that stating T implies tea. But there are situations where the object is in fact not tea (unfortunately). The first logical connective, the **not** or negation operator, is a unary operator that negates the boolean variable it is applied to, and is denoted with $\neg$. For example, a propositional sentence that states an object "is not tea" would be written:

$$\neg T$$

In the truth table of Table 1, there are 2 possible worlds.

world	T	$\neg T$
w_1	true	false
w_2	false	true

Table 1: Truth Table depicting $\neg T$

Logical conjunction, is the logical connective that represents the **and** operator, and is denoted with the symbol $\wedge$. It is a binary operator, and creates a compound sentence that outputs true if both atoms are true, and false otherwise. Using this operator, we can create sentences that state "it is tea *and* it is hot", as such:

$$T \wedge H$$

In this case, the sentence is only true if T is indeed tea and H is hot. [6]

world	T	H	$T \wedge H$
w_1	true	true	true
w_2	true	false	false
w_3	false	true	false
w_4	false	false	false

Table 2: Truth Table depicting $T \wedge H$

Similarly, there exists a connective that represents the **or** operator known as logical disjunction, and is denoted with the symbol $\vee$. If neither of these atoms are true, the sentence evaluates to false. Otherwise, the sentence evaluates to true. This operator can be used to define sentences that state "it is tea or coffee", as such:

$$T \vee C$$

Where $C(X)$ is the boolean variable representing coffee. In this case, the sentence is true if T is tea, C is coffee or both. Disjunction, together with the negation and conjunction functions, can be combined to form the remaining logical connectives.

world	T	C	$T \vee C$
w_1	true	true	true
w_2	true	false	true
w_3	false	true	true
w_4	false	false	false

Table 3: Truth Table depicting $T \vee C$

In many cases, we wish to construct sentences that select one item over another item explicitly, instead of tolerating both as the or operation does. The exclusive disjunction connective, represents the **xor** operator, and is denoted with the symbol $\oplus$. In this operation, if either atom evaluates to true, but not both, the entire sentence will evaluate to true. Should you want "tea or coffee, but not both", you can construct a sentence as follows:

$$T \oplus C$$

The sentence however, could be phrased differently, and equivalently you could state that you want "tea *and* not coffee **or** not tea *and* coffee". The propositional equivalent of this is:

$$(T \wedge \neg C) \vee (\neg T \wedge C)$$

This defines the operation of $T \oplus C$. [5]

world	T	C	$T \oplus C$
w_1	true	true	false
w_2	true	false	true
w_3	false	true	true
w_4	false	false	false

Table 4: Truth Table depicting $T \oplus C$

Atomic variables can be casual to others, the existence of one object or event may imply the existence of another. The logical connective of implication represents the **if-then** operation, denoted with the symbol $\rightarrow$. With this connective, casual connections can be made between boolean values, and sentences such as "if this is hot tea you will get burned" can be represented by the logical statement:

$$\alpha \rightarrow B$$

Where α is the sentence $H \wedge T$. Equivalently, this can be stated as the logical sentence:

$$\neg\alpha \vee B$$

In this case, the truth table is less obvious. Where the first atom is true, the case is trivial. If α is indeed hot coffee, if what follows is a burn B the expression evaluates to true, but should there be no burn the expression evaluates to false. In sentences where the first atom is false and the second atom is also false, the sentence is also trivially true. No tea implies no burn. However, w_3 is peculiar, as it evauluates to true counterintuitively. The reason is that the existence of no tea does not necessarily imply no burn. In other words, burning does not itself imply that you had tea.

world	α	B	$\alpha \rightarrow B$
w_1	true	true	true
w_2	true	false	false
w_3	false	true	true
w_4	false	false	true

Table 5: Truth Table depicting $\alpha \rightarrow C$

Finally, the last connective defined in propositional logic is known as equivalence, which represents the **equal**, or if and only if, operation. Denoted by the symbol $\leftrightarrow$, the connective implies a bi-casual relationship between two variables, and allows for sentences that describe situations such as "it is hot *if and only if* it burns". This sentence in particular is represented as:

$$H \leftrightarrow B$$

As this represents a bi-casual relationship, an equivalent statement is to say that "if its hot then you will burn **and** if you burn then it was hot", and can represented by the propositional statement:

$$H \rightarrow B \wedge B \rightarrow H$$

Equivalence is trivial to define. In cases where both atoms are the same truth value, the entire sentence should evaluate to true, and false otherwise.

world	H	B	$H \leftrightarrow B$
w_1	true	true	true
w_2	true	false	false
w_1	false	true	false
w_1	false	false	true

Table 6: Truth Table depicting $H \leftrightarrow B$

2.2 Reasoning in Propositional Logic

When given a propositional sentence, the truth value of the sentence can always be computed when supplied with the values of the atoms the sentence is comprised of. It is for this reason that propositional logic is decidable, and therefore given a knowledge base of propositional statements, any possible combination of atoms will definitively result in some value. In other words, any world is decidable.

Due to this decidability, knowledge bases can be queried by checking if a given propositional sentence is consistent with all worlds that the knowledge base expresses. This defines entailment in propositional logic, and gives means to reason with knowledge bases. However, while propositional logic is decidable, it is not easily decidable. In fact, the satisfiability of a propositional logic sentence is said to be an *NP-complete* problem [8], a problem that is inherently difficult to decide but trivial to check any given solution. This is due to the number of possible worlds growing at least 2^n, as mentioned in section 2.1.1, which is exponential in the input of n atoms.

A Propositional logic knowledge base is a 2-tuple (F, R), where F is a set of literals called facts(statements we know to be true) and R is a finite set of logical rules [11]. The rules of a knowledge base can be interpreted as a large conjunction [6]. As an example consider a knowledge base K, comprised of the following propositional sentences:

$$K = \boxed{\begin{array}{c} f_1 : tea(rooibos) \\ r_1 : hot \rightarrow burn \\ r_2 : tea \rightarrow hot \end{array}}$$

Where f denotes a fact and r denotes a rule. This knowledge base states as a fact that "rooibos is a tea", and gives 2 rules: "Hot things burn" and "tea is hot". Then, we can construct the truth table for K from the truth table for $(hot \rightarrow burn) \wedge (tea \rightarrow hot)$ as such:

$world$	hot	$burn$	tea	$K \vDash$
w_1	true	true	true	true
w_2	true	true	false	true
w_3	true	false	true	false
w_4	true	false	false	false
w_5	false	true	true	false
w_6	false	true	false	true
w_7	false	false	true	false
w_8	false	false	false	true

Table 7: Truth Table depicting $K = (hot \to burn) \wedge (tea \to hot)$

A conclusion that may be drawn from K is that since $hot \to burn$ and $tea \to hot$, it must be that $rooibos \to burn$. This inference can be made as their is at least one world where $tea \to burn$ holds. That is, $(tea \to burn) \wedge K \vDash$ true, We can confirm this from the truth table in Table 7. Precisely, $tea \to burn$ holds in worlds w_1, w_2, w_6 and w_8. We know that $tea(rooibos)$, and therefore as there is at least one world where $tea \to burn$ is true, we say that K entails this result [6], and your rooibos unfortunately will burn you. Formally:

$$K \vDash rooibos \to burn$$

3. DEFEASIBLE REASONING

Many a time, propositions are not as straightforward as one might initially believe. If propositional logic truly governed the way we think, we would not be able to make inferences that are exceptions to the normal rule. Take a penguin for example. We know that birds fly, but we know that a penguin is a bird that does not fly. Classical logic would dictate that penguins should not exist. This is of course absurd. Penguins most certainly exist (thankfully). We therefore need a mechanism to cater for such exceptions. Using the concept of defeasibility, we can extend classical logic and introduce the notion of typicality, i.e. allowing statements that say "birds *typically* fly".

Koons, Robert et al [9] describes reasoning as defeasible when an argument is rationally compelling but is not deductively valid. In other words, a statement is defeasible when the information it presents does not highlight a strict relationship, and any conclusion drawn from the statement is contextual rather than definitive.

As defeasible arguments are contingent(not necessarily false nor necessarily true), they allow us to draw conclusions from sentences that do not imply a specific answer. If for instance someone attempts to cross a street, we could infer that *typically* they would succeed in doing so. However, it is not definite. Our poor friend may be having a particularly unlucky day and get hit by a truck, and therefore would not succeed in crossing the street [9]. Hence, our original inference was defeasible.

3.1 Non-Monotonicity in Logic

Defeasibility in logic allows us to "back-track" on conclusions when presented with conflicting information. Suppose you were told by a famed zoologist that mammals do not lay eggs, and that a duck-billed platypus is a mammal. As you may be aware, a duck-billed platypus lays eggs. Therefore, in traditional logic, there can only be one conclusion. If a duck-billed platypus is an egg-laying mammal, then it simply cannot exist. We however know that duck-billed platypi do exist, and that they are indeed mammals. We therefore note that a platypus must be an exception to our original implication, and conclude that a duck-billed platypus is a mammal, hence back tracking on our original conclusion that they did not exist. This is a concept useful in autonomous systems, as it allows machines to reason presumptuously about their environment, i.e. presuming actions can be performed under certain conditions [9].

This feature of defeasibility is due to the fact that it is non-monotonic, that is upon learning new information about a system certain conclusions may be withdrawn. This is in contrast to monotonic logic, such as propositional logic, where the addition of new axioms(propositional sentences) to a knowledge base K may never decrease the conclusions that can be drawn from K [10].

3.2 Introducing Defeasiblity into Propositional Logic

The work of Kraus, Lehman and Magidor (KLM) in [14] proposed a set of natural properties of non-monotonic reasoning. In doing so a further logical connective, known as defeasible or conditional entailment, denoted with the symbol $\mid\!\sim$ was introduced to describe plausible inferences, such as the statement "if this is a mammal then typically it should not be able to lay eggs". For instance:

$$mammal \mid\!\sim \neg eggs$$

However, it is insufficient to to merely introduce conditional entailment into the language of propositional logic without first defining the semantics of the language in question.

3.2.1 Rational Consequence and the **R** *Logic*

KLM organised the essential characteristics of non-monotonic reasoning into a hierarchy of systems, ordered from worst to best according to the strength of the system. These were: Cumulative Logic **C**, loop-cumulative logic **CL** and preferential logic **P**. Preferential logic was further strengthened into KLM rational logic **R** in [13], which is the logic we shall use to define the $\mid\!\sim$ operator in this paper. For **R**, authors Lehmann and Magidor outlined 7 key properties of conditional entailment sets. Presented in the form of inference rules, the properties of logic **R** include:

Reflexivity:
Conditional inference should imply itself
$A \mid\!\sim A$

Left Logical Equivalence (LLE):
Logically equivalent formulas should entail exactly the same consequences
$\vDash A \leftrightarrow B$ then $(A \mid\!\sim C) \to (B \mid\!\sim C)$

Right Weakening (RW):
All plausible consequences that potentially exist should be accepted
$\vDash A \to B$ then $(C \mid\!\sim A) \to (C \mid\!\sim B)$

Cautious Monotonicity (CM):
Learning a new fact, the truth of which can be plausibly concluded, should not nullify previous inferences
$[(A \mid\!\sim B) \wedge (A \mid\!\sim C)] \to (A \wedge B \mid\!\sim C)$

Conjunction (And):
Conditional inference should obey propositional conjunction
$[(A \mid\!\sim B) \wedge (A \mid\!\sim C)] \to (A \mid\!\sim B \wedge C)$

Disjunction (Or):
Conditional inference should obey propositional disjunction
$[(A \mid\!\sim C) \wedge (B \mid\!\sim C)] \to (A \vee B \mid\!\sim C)$

found in [2].

Rational Monotonicity (RM):
Only additional information, the negation of which was expected, should force us to withdraw plausible conclusions previously inferred
$[(A \mathrel{|\!\sim} B) \wedge \neg(A \mathrel{|\!\sim} \neg C)] \rightarrow [(A \wedge C) \mathrel{|\!\sim} B]$

The property *CM* is a characteristic of all KLM logics, while the property *RM* distinguishes logic **R** from logic **P**. A defeasible assertion that satisfies all 7 of these properties is called a *rational consequence relation* [13]. The definition of rational consequence in **R** is an important one, as **R** appears to simulate the key characteristics of non-monotonic reasoning. Suppose for example the conditional knowledge base:

$$K = \boxed{dday \mathrel{|\!\sim} lit\_review}$$

"Normally if it is d-day then I would write my literature review"

Suppose you began to panick in loom of your inevitable literature review deadline. In **R**, $dday \mathrel{|\!\sim} lit\_review$ would not entail $dday \wedge panicking \mathrel{|\!\sim} lit\_review$ as in propositional logic, and this is a desired property of $\mathrel{|\!\sim}$. In other words, if you "normally write your literature review on d-day", it does not automatically infer that you "normally write your literature review on d-days where you are panicking" too.

3.2.2 Drawbacks of R

R is not without flaws however. We have just demonstrated that by the non-monotonicity of the $\mathrel{|\!\sim}$ operator, $\alpha \mathrel{|\!\sim} \beta$ does not entail $\alpha \wedge \gamma \mathrel{|\!\sim} \beta$. Indeed, there may be scenarios in which we would want to infer by default $\alpha \wedge \beta \mathrel{|\!\sim} \gamma$ from $\alpha \mathrel{|\!\sim} \gamma$, with the possibility to withdraw such an inference if we find it is inconsistent with the knowledge base. For example, in instances where γ is irrelevant to property β, we may tentatively infer that from $dday \mathrel{|\!\sim} lit\_review$ that $dday \wedge raining \mathrel{|\!\sim} lit\_review$ ("normally if it is d-day, even if it raining I will write my literature review"). Should we discover however that $raining$ does indeed stop me from writing my literature review, we would want to withdraw the initial inference. **R** cannot handle irrelevant information in defeasible statements, such as the example just illustrated [15].

3.3 Reasoning with a Defeasible Knowledge Base in R Logic

The Rational closure of **R** is an algorithm allows us to perform entailment over defeasible knowledge bases. It also allows us to perform reasoning with consideration to the flaw outlined in 3.2.2 .

From a high-level overview, the way this algorithm works is that each propositional sentence s_i in a knowledge base K is given a rank, denoted with a natural number. The higher the ranking of the formula, the more exceptional the formula is deemed to be [15]. Formally, this rank is known as a ranked interpretation R, and is defined as pair $\langle V, \prec \rangle$, where V is a subset of all the possible valuations of K and $\prec$ is a unique ordering of all of s_i. Lehmann and Magidor showed that there exists a minimal R for any K, and therefore each of s_i is interpreted as normally as possible. That is, there some R that produces a minimal ordering of $\prec_s$.

Given this, a defeasible sentence $\phi = A \mathrel{|\!\sim} B$ is said to be in the *rational closure* of K should the ranked interpretation of A be strictly less than the ranked interpretation of $A \wedge \neg B$ or where A has no rank [12] [18]. The exact steps of this algorithm and the minimisation of R are outlined by Casini and Straccia, and can be found in [2].

3.3.1 An illustration of Rational Closure

Suppose the following (classic) example:

$$K = \boxed{\begin{array}{c} f_1 : penguin(nemo) \\ r_1 : bird \mathrel{|\!\sim} fly \\ r_2 : penguin \rightarrow bird \\ r_3 : penguin \mathrel{|\!\sim} \neg fly \end{array}}$$

Where f_1 is a fact, r_1 and r_3 are defeasible rules and r_2 is a strict rule. There is an obvious conflict. It is stated that birds typically fly and a penguin is a bird. However it is also stated that penguins typically do not fly, leading to a (potential) contradiction. Specifically, nemo cannot exist unless the contradicting statement is overruled. Using the semantics of rational closure, the minimal ranked interpretation R of K is given by:

$$R = \begin{array}{|c|c||c|} \hline \text{"least typical"} & 3 & r_2 : penguin \rightarrow bird \\ \uparrow & 2 & r_3 : penguin \mathrel{|\!\sim} \neg fly \\ \text{"most typical"} & 1 & r_1 : bird \mathrel{|\!\sim} fly \\ \hline \end{array}$$
$$\prec = (r_1, r_3, r_2)$$

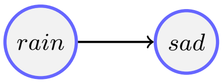
As we now have R, we have the ability to query K. Perhaps you want to check "if something flies is it typically not a penguin?". Formally, $K \models ?fly \mathrel{|\!\sim} \neg penguin$. In this case, we need to check that the rank of fly is less than the rank of $fly \wedge \neg(\neg penguin)$. As it happens, $\mathrel{|\!\sim} fly$ sits at rank 1, where as $fly \wedge penguin$, which can be inferred from r_2, sits at rank 3. Therefore, $fly \mathrel{|\!\sim} \neg penguin$ is in the rational closure of K and we deduce that:

$$K \models fly \mathrel{|\!\sim} \neg penguin.$$

4. BAYESIAN NETWORKS

Formal logic systems are not the sole way to model real-world phenomena. In fact, there exist more numerical ways of viewing the world, as opposed to a simple binary system as in propositional logic. One such way is through the lens of probability, which is a method of quantifying the likelihood of events occurring. However, the way we calculate the probability of an event happening in a system is not straightforward. Certain events in the system may be dependent on others, and to calculate the probability of some event occuring one must consider the probability of the dependencies of that event as well.

A Bayesian Network is a visual construct that allows us to represent such dependencies between probabilistic events in a structured manner. It consists of a graph(a set of nodes and edges connecting those nodes), depicting the relationships between random variables, and a set of probabilities associtated with those variables [3]. The random variables in a network are boolean in nature. They are either true or false. Suppose we wanted to model the way rain interacts with mood, in particular being sad. We would define two random variables, $rain$ which represents if it is raining, and sad which represents if we are sad. If we know that the rain makes us sad, this could represented by the following Bayesian Network:



$rain$	$P(rain)$
true	0.1
false	0.9

$rain$	sad	$P(sad\|rain)$
true	true	0.6
true	false	0.5
false	true	0.4
false	false	0.8

Figure 1: A Bayesian Network and its CPTs

The construction is formatted in such a way that making inferences about an event in a system arranged as a Bayesian Network is quite efficient. This is due to a variety of probabilistic reasoning algorithms being developed that take advantage of the simple structure of a Bayesian Network. Before we formally define a Bayesian Network, it is necessary to go over the necessary prerequisite probability theory. We do however assume some familiarity with basic probability theory.

4.0.1 Bayes Rule

Bayesian Networks rely on **Bayes Rule**, which describes method of revising probabilities in the light of new information [22]. This is known as *conditional probability*, and is defined accordingly:

Let A and B be two events in some sample space S. Then, the conditional probability of event B given that event A has occurred is denoted with $P(B|A)$, is given by:

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

In other words, the probability of B given A is the probability of A and B occurring divided by the probability of A occurring.

This rule can restructured, as we know that $P(A, B) = P(A|B) \cdot P(B)$ from the definition, giving us:

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

Using this formula, we can calculate the probability of any event A given any number of conditions B_i as $P(B|A_1, A_2, ..., A_n)$

4.0.2 Probability distributions

A **probability distribution** is a statistical function that maps all the possible values that random variables in a system can take to some probability [22]. For example suppose the system describing $rain$ and sad. Both $rain$ and sad can each take one of two values, true or false, leading to four potential states the system can take:

$rain$ **and** sad
$rain$ **and** $not\ sad$
$not\ rain$ **and** sad
$not\ rain$ **and** $not\ sad$

A probability distribution will then map each state to a certain probability, such that if you query the distribution with some state of the system it will provide the probability associated with that state.

4.0.3 The Structure of a Bayesian Network

The content of the following section(5.0.3) cites Darwiche, A et al [6]

Formally, a Bayesian Network BN is pair $\langle DAG, CPT \rangle$, where DAG is a directed acyclic graph and CPT is a set of conditional probability tables. DAG nodes represent random variables, with the edges that connect those nodes representing a dependency relation. Variables are deemed *independent* of each other should no edge directly connect them. The CPT of BN specifies probability distribution of each of the variables and their respective parents in the modelled system. There can only ever be one such CPT specifying a network, and as such any Bayesian Network BN has completeness and consistency guarantees.

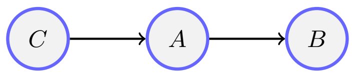
The power in Bayesian Networks relies on the notion of variable independence. Given a some variable x in BN, if we know the values of the parents of x, then x is also independent from its children. Due to this independence, in any computation only probabilities associated with the variable in question need to be considered, drastically decreasing computation time.

4.1 Bayesian Reasoning

A Bayesian Network would be useless construct should we not be able to reason with the structure for information. There exist several types of queries that can be used to reason with a Bayesian Network, outlined by Darwiche in [6]. For the purposes of this paper, we shall focus on the most obvious query available in a BN, i.e. the probability-of-evidence query, which allows us to query the network for the probability of certain events occurring in the system. Given one or many evidence variables x_i, the query $P(x_1, .., x_n)$ will output the probability associated with $x_1, .., x_n$. This is done by use of an algorithm called *Factor Elimination*, which sequentially removes variables that are not associated with the query, eliminating the need to calculate the probability over all the variables in the network. The details of this algorithm are covered by Darwich [6].

5. LOGICAL IMPLICATION IN BAYESIAN NETWORKS

Ultimately, we seek an approach to Bayesian reasoning in which we combine logical causality with a Bayesian Network in question in hopes of further simplifying the said network. This can be done as the nodes in a network are in reality propositional atoms holding either a true or false value, and any dependencies depicted in the network could be depicted as logical dependencies, albeit with a certain probability attached to the relationship. For example, suppose the following Bayesian Network:



$$P(A, B, C) = P(C) \cdot P(A|C) \cdot P(B|A)$$

Figure 2: Bayesian Network of A, B and C

You may notice that at a glance there seems to be a link between logic statements and such a network. Specifically, we propose that in non-monotonic logic this could be interpreted as the conditional entailment of propositional atoms A and B as such:

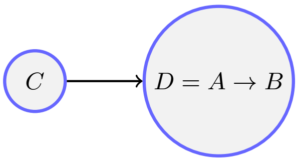
$$A \mathrel{|\!\sim} B$$

This construction would allow us to specify additional dependencies within a Bayesian Network in the form of logical implications between atoms, such as $C \rightarrow B$, which can be done as

Bayesian Network variables are largely independent from one another. In attempting to develop such a model, we will look at how this construction affects the structure of the network and its associated probabilstics, and how to reason with it.

5.0.1 *Classical Implication*

Initially we aim to examine the consequence of a propositional implication between Bayesian Network variables. Given such variables A, B and C from Figure 2 , if $A \rightarrow B$ our initial suspicion is that $P(B|A) = 1$. Indeed, this would mean that variables A and B would collapse into some single variable D, where $P(D) = P(A|C)$.



$$P(C, D) = P(C) \cdot P(A|C) \cdot (P(B|A) = 1)$$

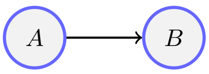
Figure 3: Bayesian Network of $C, A \rightarrow B$

5.0.2 *Defeasible Implication*

With the addition of defeasiblity, the construct becomes more complex, as if $A \mid\!\sim B$ we cannot deduce that $P(B|A) = 1$ as in 5.0.1 . We note however that if $P(B|A) > 0$ and $P(B|A) < 1$, then it may imply that $A \mid\!\sim B$.

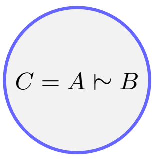
If this is the case, while rational closure is a suitable reasoning mechanism in non-monotonic logic, we are unable to deduce what the effect of ranked interpretations of variables would be on a Bayesian Network system. It is not obvious what the notion of typicality means in Bayesian Network system. Nevertheless, in examining a combined mechanism we will attempt to formalise the relationship between conditional entailment and conditional probability, ultimately with a formalism that resembles the follwing:

If A and B are variables in the Bayesian Network β such that:



$$P(A, B) = P(A) \cdot P(B|A)$$

Then, if $A \mid\!\sim B$ $\exists$ a sequence of probabilities p where $\forall q \epsilon p, 0 < q < 1$, and q is the conditional probability corresponding to a non-monotonic propositional sentence in the minimal ranked interpretation R of some Knowledge Base K, where $K \vDash A \mid\!\sim B$. Suppose a function ξ, that selects the most typical $q \epsilon p$ given some logical rule $\phi \epsilon V(R)$ of the form $\phi = x \rightarrow y$ or $\phi = x \mid\!\sim y$. Hence, if $A \mid\!\sim B$, then β becomes:



$$P(C) = \xi(p, \phi)$$

Essentially, to calculate the probability of this new node C, we wish to assign probabilities to the rankings in R, and given some logical formula ϕ we will determine how "typical" ϕ is according the rankings and output the associated probability.

5.1 Alternative Solutions

The literature on integrating propositional logic into Bayesian networks is well established. Pearl describes various mechanisms for probabilstic reasoning in intelligent systems in his book [21], and the work of Cozman and Mauá et al 2016 [20] describe how a class of logics, known as Description Logics, can be used to represent Bayesian Networks.

There may be other ways to deal with the notion of $A \mid\!\sim B$ in a Bayesian Network. Most notably is the construct of fuzzy sets and degrees of truth in Fuzzy Logic(a logic in which logical sentences are assigned probabilties of occuring), which may aid in determining the degree of typicality implied by a non-monotonic statement such as $A \mid\!\sim B$[16]. Furthermore is the idea of non-monotonicity in fuzzy logic, as described by Castro, Trillas and Zurita et al 1995 [19], which can be used to deal with exceptions in much the same way as propositional non-monotonic logic.

Other probabilistic logic systems do exist however. Probabilistic Theorem Proving as described by Kautz and Singla et al 2016 [17] allows for reasoning with uncertainity proving theorems in first order logic, as well as checking the satisfiability of classical systems. Plausible probabilistic reasoning systems are described

6. CONCLUSIONS

In this literature review, we concentrated on the properties of logical reasoning systems, and how feasible an integration with Bayesian Networks and these systems would be. As we have seen from the literature, both classical and non-monotonic logic are well established logic formalisms for reasoning. Specifically, they are able to represent casual and typical relationships between variables, and do so in an efficient manner.

The literature on Bayesian reasoning underlines the explicit structure of Bayesian Networks. In particular, we have identified that the most valuable aspect of a Bayesian Network is the notion of independence between variables that are not directly linked. This specific property allows the probabilities in a Bayesian Network to be calculated in a straightforward manner, where only the parents of the node in question need to be considered for the calculation.

From an initial point of view, the fields of logic and Bayesian reasoning appear to have obvious synergy. In particular, the intersection of propositional logic and Bayesian Networks appear to provide an obvious mechanism of shrinking Bayesian Networks, and therefore their associated complexity. However the same cannnot be said of the intersection of non-monotonic logic and Bayesian reasoning, as there a few glaring issues. Foremost, there is no obvious way to correlate typicality with conditional probability. There is also no formalism in the literature deals with defeasiblity in Bayesian Networks. Indeed, there is little to no literature on the integration of such a defeasible reasoner with bayesian network either, and as such we conclude these topics will need to be further explored.

We propose that union of these two reasoning mechanisms would allow us to essentially shrink the size of any given Bayesian Network by supplementing the network with a logical knowledge base. This would be done by viewing the variables in the network as propositional atoms, and then using the knowledge base to draw causality between these atoms. We deduce that in drawing a logical implication between these variables that they combine into a single variable with a single probability.

Overall, we have determined a key area in the intersection of non-monotonic logic and Bayesian Networks that presents a gap in the literature. We conclude that further research is required into the role of logic and defeasibility in Bayesian Network reasoners, as the theory behind such a combined system is largely underdeveloped and the benefits of which are unknown.

7. REFERENCES

[1] Hunter, A. (2002, July 1). Propositional Logic: Deduction and Algorithms (Book Review). Studia Logica: An International Journal for Symbolic Logic. Kluwer Academic Publishers.

[2] Casini G., Straccia U. (2010) Rational Closure for Defeasible Description Logics. In: Janhunen T., Niemelä I. (eds) Logics in Artificial Intelligence. JELIA 2010. Lecture Notes in Computer Science, vol 6341. Springer, Berlin, Heidelberg

[3] Darwiche, A. (2010). Bayesian networks. Communications of the ACM, 53(12), 80–90. doi:10.1145/1859204.1859227

[4] Delancey, C. (2017). A concise introduction to logic. Geneseo, NY: Published by Open SUNY Textbooks, Milne Library, State University of New York at Geneseo.

[5] Ben-Ari, M. (2012). Mathematical logic for computer science. Springer Science & Business Media.

[6] Darwiche, A. Modeling and Reasoning with Bayesian Networks. Cambridge University Press, Cambridge, 2009.

[7] Eiter, T., Gottlob, G. (2006). Reasoning under minimal upper bounds in propositional logic. Theoretical Computer Science, 369(1), 82–115. doi:10.1016/j.tcs.2006.07.054

[8] Sipser, M. (2006). Introduction to the Theory of Computation (Vol. 2). Boston: Thomson Course Technology.

[9] Koons, & Robert, (2005) Defeasible Reasoning, The Stanford Encyclopedia of Philosophy (Winter 2017 Edition), Edward N. Zalta (ed.)

[10] Bundy A., Wallen L. (1984) Non-Monotonic Reasoning. In: Bundy A., Wallen L. (eds) Catalogue of Artificial Intelligence Tools. Symbolic Computation (Artificial Intelligence). Springer, Berlin, Heidelberg

[11] Antoniou, G., Dimaresis, N., & Governatori, G. (2008, March). A system for modal and deontic defeasible reasoning. In Proceedings of the 2008 ACM symposium on Applied computing (pp. 2261-2265). ACM.

[12] Strasser, Christian & Antonelli, G. Aldo, (2001) Non-monotonic Logic, The Stanford Encyclopedia of Philosophy (Summer 2018 Edition), Edward N. Zalta (ed.)

[13] Lehmann, D., & Magidor, M. (1992). What does a conditional knowledge base entail?. Artificial intelligence, 55(1), 1-60.

[14] Kraus, S., Lehmann, D., & Magidor, M. (1990). Nonmonotonic reasoning, preferential models and cumulative logics. Artificial intelligence, 44(1-2), 167-207.

[15] Giordano, L., Gliozzi, V., Olivetti, N., & Pozzato, G. L. (2013). A Semantics for Rational Closure: Preliminary Results. In CILC (pp. 99-113).

[16] Bergmann, M. (2008). An introduction to many-valued and fuzzy logic: semantics, algebras, and derivation systems. Cambridge University Press.

[17] Kautz, H., Singla, P. (2016). Technical Perspective: Combining logic and probability. Communications of the ACM, 59(7), 106–106. doi:10.1145/2936724

[18] Booth, R., Casini, G., Meyer, T. A., & Varzinczak, I. J. (2015, June). On the Entailment Problem for a Logic of Typicality. In IJCAI (pp. 2805-2811).

[19] Castro, J. L., Trillas, E., & Zurita, J. M. (1998). Non-monotonic fuzzy reasoning. Fuzzy Sets and Systems, 94(2), 217-225.

[20] Cozman, F., & Mauá, D. (2016). The Complexity of Bayesian Networks Specified by Propositional and Relational Languages.

[21] Pearl, J. (2014). Probabilistic reasoning in intelligent systems: networks of plausible inference. Elsevier.

[22] Underhill, L., Bradfield D. (2014). INTROSTAT, Department of Statistical Sciences, University of Cape Town