Department of Infomedia

Master Thesis

On Common Beliefs

By:Vemund Ytrehus
Supervisor: Thomas Ågotnes

June 1, 2019

Abstract

This thesis discusses how probabilistic multi-agent common belief can be defined. It also attempts to provide insight into what constitutes belief. Further, it proposes an algorithm to measure the strength of such common belief by redefining common belief as a bayesian network in form of a directed cyclic graph, and discusses common belief in terms of computational complexity and the depth of belief.

Contents

List of Figures

Content Page Reference

List of Definitions:

List of Propositions:

to common belief, only $AB1$.

List of Claims:

List of Algorithms:

Chapter 1

Introduction

This thesis focuses on *probabilistic belief* in a multi-agent system. More specifically, it focuses on common belief. Common belief is the belief version of common knowledge, which in turn is knowledge that is known by everyone, that is known by everyone to be know by everyone, and so on.

This can be important for deducing on behalf of an agent whether or not to act, in a setting in which group belief affects how others act. For example, an agent deducing whether or not to buy stocks, would benefit from such knowledge. Being able to deduce based on probabilistic belief is essentially the same thing as knowledge, but with added advantages.

Further, numerous well known examples illustrate situations in which knowledge of what others know is necessary for success. The next section looks at some well known dilemmas or puzzles that shows different aspects of the issues discussed in this thesis.

1.1 Motivation

The examples here are well known puzzles and problems related to common knowledge. They can somewhat simply be cast onto the realm of belief, by introducing uncertainty into them, e.g. by saying an agent can not see well and may therefore be mistaken in his observation, or changes like that.

1.1.1 Example 1: Prisoner's dilemma in game theory

Game theory [1] is a field of research which tries to explain how agents act in situations where other agents may interfere with their own outcome. The theory deals with how agents may behave in ways that may be bad for both the actor and all other individuals.

The Prisoner's Dilemma is one of the most well known problems from game theory. It is defined in [2] as: Two crooks, *agent1* and *agent2*, are arrested and face trial. The prosecutors do not have sufficient evidence to convict on the more serious charges, so each prisoner is independently offered a one-off deal of testifying in secret against the other, in exchange for being released free of charges, so that the one who is being testified against will get a much harsher sentence. However, if *both* prisoners accept the deal and testifies, both prisoners will be convicted. The rules are specified for the prisoners as follows:

- If *agent1* and *agent2* each decide to testify against the other, each of them will be sentenced to two years in prison.
- If *agent1* takes the deal and testifies against *agent2*, but *agent2* remains silent, then *agent1* will go free and *agent2* will serve three years in prison (and vice versa). It is assumed that the betrayed prisoner will not learn of the betrayal, or will not be able to take revenge in any way.
- If neither *agent1* nor *agent2* testifies, both of them will only serve one year in prison.

We see that the *socially optimal* outcome is that both testify, with a mean of 1 year of prison per crook. If both testify, the mean is 2 years of prison. If only one testify, then the mean outcome is 1.5 year of prison. But since the rewards for betrayal outweighs the rewards of collaborating, we tend to end up with the case that both testify and accept the deal.

The game theoretic perspective is that each agent is governed only by self-interest, but it ignores aspects such as trust, loyalty and such aspects. In real life, such concepts can have great importance, as we would often see that prisoners will not betray each other due to reasons such as loyalty, trust, or even fear of repercussions. Since trust and loyalty sometimes happens and sometimes do not, a framework for representing the problem must include some way to include it. Looking at such concepts as a probabilistic belief, is one way to handle it.

If trust, or belief, is included into the solving of the problem, then the agents may reason in a way so that the socially optimal outcome is reached.

1.1.2 Example 2: Two Army Problem

Also known as the Two Generals' Problem, the Two Generals Paradox, and the Coordinated Attack Problem, this problem is described in [3] as:

> *Two armies, each led by a different general, are preparing to attack a fortified city. The armies are encamped near the city, each in its own valley. A third valley separates the two hills, and the only way for the two generals to communicate is by sending messengers through the valley. Unfortunately, the valley is occupied by the city's defenders and there's a chance that any given messenger sent through the valley will be captured. While the two generals have agreed that they will attack, they haven't agreed upon a time for attack. It is required that the two generals have their armies attack the city at the same time in order to succeed, else the lone attacker army will die trying. They must thus communicate with each other to decide on a time to attack and to agree to attack at that time, and each general must know that the other general knows that they have agreed to the attack plan. Because acknowledgement of message receipt can be lost as easily as the original message, a potentially infinite series of messages is required to come to consensus.*

The two army problem is often used as an example that knowledge can be hard to establish. Even if one of the sides receives a message that they should attack, they will not know that the sender knows that the message is received. This is an example of a situation in which you need to establish *common knowledge* before you can act. The problem here, which makes it a well known problem, is that you can never have common knowledge of an attack plan given this setup, as there is no closure in which everyone knows that everyone knows that, etc. Such a closure would be infinite. If we assign the messengers a distinct probability for their runs, this would illustrate how common (probabilistic) belief could operate. This example shows that the message sending back and forth can not be infinite but must be trimmed at some amount of backs and forths.

1.1.3 Example 3: A competitive hat puzzle

A simple *competitive hat puzzle* is described in [4] as follows:

> *Three players are told that each of them will receive either a red hat or a blue hat. They are to raise their hands if they see a red hat on another player as they stand in a circle facing each other. The first to guess the colour of his or her hat correctly wins. All three players raise their hands. After the*

players have seen each other for a few minutes without guessing, one player announces "Red", and wins. How did the winner do it, and what is the color of everyone's hats?

Solution:

A brute-force approach goes like this. Consider what player 1 observes. hat_2 and hat_3 are the colours of the hats of players 2 and 3, $hand_2$ and $hand_3$ represent the states of the hands of players 2 and 3, respectively.

hat_1	hat_2	hat_3	$hand_2$	$hand_3$
red	*red*	*red*	*raised*	*raised*
blue	*red*	*red*	*raised*	*raised*
red	*blue*	*red*	*raised*	*raised*
blue	*blue*	*red*	*raised*	*notraised*
red	*red*	*blue*	*raised*	*red*
blue	*red*	*blue*	*notraised*	*raised*
red	*blue*	*blue*	*raised*	*raised*
blue	*blue*	*blue*	*notraised*	*notraised*

For each distinct set of observations except one, player 1 can distinguish her own hat colour. But if player 1 observes only *red* hats and (in consequence) only *raised* hands, player 1's hat colour.
However, the other players can perform a similar reasoning. If player 1's hat is *blue*, both player 2 and player 3 will be able to distinguish their own hat colour. Hence, if "a while" has passed without players 2 or 3 calling their own hat colour, player 1 can conclude that her hat colour is *red*.

This example can also be be cast onto a probabilistic space, by for example saying that the agents can see the raised hands with only a certain probability, or that some agents are colorblind.

1.1.4 Example 4: A collaborative hat puzzle

The same Wikipedia article [4] also describes a simple *collaborative hat puzzle*:

According to the story, four prisoners are arrested for a crime, but the jail is full and the jailer has nowhere to put them. He eventually comes up with the solution of giving them a puzzle so if they succeed they can go free but if they fail they are executed. The jailer seats three of the men into a line. B faces

the wall, C faces B, and D faces C and B. The fourth man, A, is put behind a screen (or in a separate room). The jailer gives all four men party hats. He explains that there are two black hats and two white hats, that each prisoner is wearing one of the hats, and that each of the prisoners see only the hats in front of him but neither on himself nor behind him. The fourth man behind the screen can't see or be seen by any other prisoner. No communication among the prisoners is allowed. If any prisoner can figure out what color hat he has on his own head with 100 % certainty (without guessing) and tell the jailer, all four prisoners go free. If any prisoner suggests an incorrect answer, all four prisoners are executed. The puzzle is to find how the prisoners can escape.

Solution:

[4] gives the solution as follows: *The prisoners know that there are only two hats of each color. So if D observes that B and C have hats of the same color, D would deduce that his own hat is the opposite color. However, if B and C have hats of different colors, then D can say nothing. The key is that prisoner C, after allowing an appropriate interval, and knowing what D would do, can deduce that if D says nothing the hats on B and C must be different; able to see B's hat, he can deduce his own hat color.*
(In other words, A and B does not really play any role in the game, except from having heads to put hats on. Also, it can be argued that *silence* for a certain amount of time is a way of using a timing channel to pass information.)

We can also here cast the problem onto a probability space, so that some agents for example can not see color accurately.

1.2 Contributions of this thesis

This thesis contributes with the introduction of an algorithm to calculate the strength of probabilistic common belief. It also contributes with arguments to limit the depth at which common belief is calculated, attempts to better define a modal operator for multilevel probabilistic belief, and explore complexity considerations and other aspects with regards to common belief.

1.3 Overview

The rest of this thesis is structured as follows:

Chapter 2 goes through the theoretical background required for the thesis.

Chapter 3 discusses and defines logical belief.

Chapter 4 mentions how my work can be expanded.

Chapter 5 goes through some thoughts that occurred while writing the thesis such as potential problems.

Chapter 6 gives a conclusion.

Chapter 2

Terminology and Background

For reference, the following table contains a list of some of the notation that will be used throughout the thesis.

Term	Meaning
$\neg$	Logical not
$\bigvee$	Logical or
$\bigwedge$	Logical and
$\Rightarrow$	Logical implication
$\Leftrightarrow$ / iff	If and only if
$\exists$	Existential quantifier
$\forall$	Universal quantifier
$K_a\phi$	Epistemic operator
$B_{a,d}\phi$	Belief operator
$a \vDash b$	b is valid in a
$f_X(X = x_i)$ or Pr_i	probability mass functions (used interchangeably for convenience)

2.1 Formal logic

This section gives an overview of basic formal logic. See, for example [5].

When holding certain information, we can often deduce other pieces of information. For instance, let us say that we are privy to the following information:

A: We are in the city of Bergen. B: It does always rain in the city of Bergen.

From this, we can intuitively deduce the following:

C: It is currently raining where we are.

Formal logic denotes formal systems for reasoning and deduction based on information, to formally make those deductions which was done informally when reasoning about C given A and B. These systems can be divided into two subsystems: Propositional logic, and predicate logic, the latter which is an extended version of the former.

2.1.1 Propositional Logic

Propositional logic deals with pieces of information which either is, or is not. A proposition is a 1-bit piece of information which can be contained in one bit of information, which is called a Boolean value. The proposition has whatever meaning we ascribe to it. For example, we can have a proposition we just call p, which holds the meaning of whether or not it is currently raining. If it is, then the value of p is true (or, 1), while if it is not, the value of p is false (or 0).

We can bind these values together by means of simple, deductive operations. In practice, all such operations can be constructed as a conjunction of three basic operations: $\neg$, $\bigvee$, and $\bigwedge$. In fact, either of the last two ones are enough, since both can be constructed with $\neg$ and the other by means of De Morgan's laws, which states that $\neg(a \bigvee b) = \neg a \bigwedge \neg b$ and that $\neg(a \bigwedge b) = \neg a \bigvee \neg b$.

- $\neg$: This is the negation operator in propositional logic. In essence, it says that the proposition coming after it has the opposite value of that which it had previous to the operator.
- $\bigvee$: This is a disjunction operator which binds together the value on each side of it, in essence saying that if either side is true, the whole is true.
- $\bigwedge$: This is a conjunction operator which binds together the value on each side of it, but in a different way than the or-operator. If both sides are true, and only if both sides are true, the total value will be true.

We can then use this propositional logic to make more complex propositions. For example, if we use the example given above of p denoting whether or not it rains, we can make the following claim:

$p \bigvee \neg p$

This simply means that either p is true, or p is false. Since p is the propositional value of whether or not it is raining, what this means is simply the notion that either it rains, or it does not.

Another important notion, is the notion of implication: If p then q. For example, we might want to say that if it is raining, there is no party tomorrow: if p, then q, where p denotes "it is raining" and q denotes "there is no party tomorrow". This can be done by combining the disjunction and negation from above, and simply say:

$\neg$ p $\bigvee$ q

Now this is rather annoying to read, as it says that either it is not raining or there is no party tomorrow. But this is logically the same thing as saying that if it is raining, there is no party tomorrow. We intuitively understand the latter better than the former, and as such we would like an operator so that we can easily write it. We do this with the the $\Rightarrow$, the implication operator in propositional logic, so that p $\Rightarrow$ q says the same thing as the statement above.

We will also often use the logical implication both ways. In this case, I will use $\Leftrightarrow$ to denote a bidirectional implication, which in effect has the meaning of if and only if. For example, we can then express the following tautology: $p \Leftrightarrow p$. This simply means that p is true, if and only if p is true.

A logical formula which is valid, is a formula which is true no matter which the valuation of the propositions in the formula. For example, $p \bigvee \neg p$ is a formula which is valid, because no matter what value the proposition p has, the value of the entire formula is true. $a \bigvee \neg b$ is not a valid formula, because depending on the value of a and b, the total value of the formula may be true or false. Similarly, an argument can be said to be valid, if every argument of that form necessarily is true.

With the operators introduced, we can take a look back at the previous section's statements, where:

- A: We are in the city of Bergen.
- B: It does always rain in the city of Bergen.
- C: It is currently raining where we are.

This argument can then be formally represented with propositional logic as the following, simple propositional argument:

(A $\bigwedge$ B)$\Rightarrow$C.

Notice that in using propositional logic here we have ignored some of the finely grained information we had, because we could not express it. "We are in the city of Bergen.", "It does always rain in the city of Bergen.", "It is currently raining where we are." are all atomic particles of the expression above, while they are each more complex than what can be represented by a simple value of true or false. For example, "We are in the city of Bergen." suggests that there is an actor, or agent we are referring to (namely "We"), and "It does always rain in the city of Bergen." suggests a temporal aspect of "always" as opposed to just now or the likes. Because propositional logic deals with simple, Boolean values which can either be true or not true, a lot of information is simply ignored when formalizing a logical reasoning. Predicate logic tries to handle some of these problems by more finely graining the information.

2.1.2 Predicate Logic

Predicate logic builds on propositional logic, but infuses it with the concepts of a predicate and of quantifiers.

A predicate is a function, $P(x)$, which gives you a true or false value based on the predicate variable x (or multiple predicate variables). In essence, then, a predicate is a function which for any given input will give you a proposition (a true or false value).

For example, a simple predicate giving you the value true for all positive numbers (0-exclusive) can be defined as $P(x \in \mathbb{R}) = x > 0$. As a side note, in pursuit of simplicity, I will often leave the set membership of the variables to be left implicit unless there is a reason not to do so, so the above will rather be written $P(x) = x > 0$ and it will be left implicit that x is a real number, and not, say, a day of the week.

A quantifier is a construct allowing you to define truth values about a group of items. There are two quantifiers: The existential quantifier, and the universal quantifier.

An existential quantifier, $\exists$, contains a variable and a formula, and means that for said variable, the formula following it is true for some value which the variable can hold. That is, the existential quantifier over the quantified variable holds as true if and only if there exists some value for the variable in the set the variable is an element of, for which the formula holds true. For example, $\exists x \in \mathbb{N}(x > 5)$ means that there exists some natural number which is greater than 5.

Similarly, a universal quantifier, $\forall$, contains a variable and a formula, and means that for said variable, the formula following it is true for all values the variable can hold. A formula in which a universal quantifier is the outermost part is true, if no matter which value is assigned to the variable the formula holds true. For example, $\forall x \in \mathbb{N}(x \geq 0)$ is a true formula, simply stating that all natural numbers are greater or equal to 0.

Going back to our previous example with:

- A: We are in the city of Bergen.
- B: It does always rain in the city of Bergen.
- C: It is currently raining.

Using predicate logic we can now formalize this more accurately. Notice that there is strictly speaking no need to clarify B with a universal quantifier, but we will do that here anyway as an example.

Let P(x) be a predicate which is true if a is Bergen. Then

- A: $P(a)$
- B: $\forall x(P(a) \Rightarrow p)$
- C: p

We can see that C is a valid conclusion of A and B, since given A and B, C must necessarily be true.

2.2 Multi-agent Epistemic logic

This Section is based on [6].

2.2.1 Definition of Multi-agent Epistemic Logic

Epistemic logic is the logic for reasoning about knowledge. It is a modal logic, meaning that it incorporates modalities into the logical language, such as possibility or necessity or the like. In the case of epistemic logic, the modality incorporated into the language is that of knowledge. Epistemic logic builds on propositional logic, but the epistemic operators are also similar to predicates from predicate logic. That it is multi-agent means that it is able to describe and deduce the knowledge of more than one actor in a system.

When defining a modal language, we need to have a set of facts, a set of modal operators, and a set of agents. The set of facts is simply the set of propositions from

propositional logic, which can be true or false. In this case, let $Prop$ be the set of primitive propositions. The set of agents are all the actors which we want to describe as having knowledge or belief within a setting. Let Ag be the set of all agents. For instance, if all our agents are $Alice, Bob$, and Eve, then $Ag = \{Alice, Bob, Eve\}$. Finally, the set of operators that are the modal operators in the language, is the set of operators which exists in the language to define modal operations. In the context of a multi-agent system, we need a set of operators that is the set of binary relations $\{K_a g\}$, where K is the modal operator to express knowledge possessed by an agent $ag \in Ag$.

Epistemic logic with multiple agents can then be defined by the Backus normal form in equation (2.1):

$$\phi := p \in Prop \mid \neg\phi \mid \phi_1 \bigwedge \phi_2 \mid K_a\phi \tag{2.1}$$

In addition, $\bigvee$, $\Rightarrow$, and $\Leftrightarrow$ are implicitly included in the fact that they can be derived logically from $\neg$ and $\bigwedge$.

Using the logic defined here, we can now easily denote any agents' knowledge by using the modal operator K, and further reason semantically with these operators.

2.2.2 Semantics of Epistemic Logic

In propositional logic and predicate logic the semantic values of a given formula is usually quite straight forwardly defined as true or false. With the epistemic operators, this is made somewhat more difficult, as what an agent knows depends on the agent in question.

We can think of the semantics of the epistemic operator in the following way: Each agent sees the world as having one or more possible **state**, depending on what that agent knows. If an agent sees a value as having one of two possible actual values, but the agent does not know which one, that agent considers both those worlds, or states, possible. For example, if an agent a is sitting in an underground shelter, with no information about the weather outside, he can consider both the case that it is raining outside or that it is not raining outside as possible cases. If we denote the proposition of "it is raining" as p, then as long as $\neg K_a p$ and $\neg K_a \neg p$ both hold true a considers both states as possibly true. If either $K_a p$ or $K_a \neg p$ is true, then that agent will consider only the states in which either p or $\neg p$ holds true to be possible states.

The concept of possible states can be formalized in a **Kripke model**. A Kripke model is a representation of the semantics within a non-classical logic, such as a modal logic. The Kripke model includes three sets: A set of states, or worlds, a set of relationships between those states, and a set of evaluations for each atomic proposition in each state. Formally, a Kripke model M for epistemic logic is definable as:

$$M = (S, R, V)$$

where:

- S is the set of all possible states.
- R is the set of accessibility relations between the states in S for each agent $a \in Ag$. An accessibility relation (s, t) denotes that a sees state t as a possible true state if a is in state s. We let R_a denote the accessibility relation for agent a. If there is a relation between the states s and t, we say that t is accessible from s. While each accessibility relation is a binary relation, you can alternatively see it as consisting of functions $R_a(s)$, giving all states reachable from s.
- V is the set of evaluations to true or false for each atomic proposition in each state S. $V(s \in S)(p \in Prop)$ describes the value of the proposition p in state s.

When disregarding the evaluation set we can also be talking about a Kripke frame $F = (S, R)$.

Furthermore, we will very often talk about a given model of a given state, M and s. We write this simply as M, s. As a shorthand for validity in a model, we use $\vDash$, simply meaning that the right hand side is valid given the left hand side.

The semantics of a formula in epistemic logic can then be drawn from the Kripke model as follows. Given $p \in Prop$ and the Kripke model $M(S, R, V)$,

$$M, s \vDash p \text{ if and only if } V(s)(p) \text{ is true.}$$
$$M, s \vDash \neg p \text{ if and only if not } M, s \vDash p.$$
$$M, s \vDash p \bigwedge q \text{ if and only if both } M, s \vDash p \text{ and } M, s \vDash q.$$
$$M, s \vDash K_a p \text{ if and only if in every state } t \text{ that is accessible from } s \text{ for the agent } a, M, s \vDash p.$$

As an example, say that we have the following facts:

- It is not raining.
- *Alice* does not know if it is raining.

- *Bob* knows that it is not raining.

- *Eve* knows that *Alice* does not know that it is raining, but does not know that *Bob* knows that it is raining.

This can now be formalized, using the epistemic logic defined, as:

Agents = $\{a, b, e\}$

- $\neg p$

- $\neg K_a(\neg p) \bigwedge \neg K_a(p)$

- $K_b(\neg p)$

- $K_e(\neg K_a(\neg p) \bigwedge \neg K_a(p)) \bigwedge \neg K_e(K_b(\neg p))$

2.2.3 Group Knowledge

Since we are dealing with groups of agents, where each agent may share the same knowledge with other agents or have different knowledge from other agents, it is also natural to speak about group knowledge. When talking about group knowledge, there are three major kinds of group knowledge.

One kind of group knowledge is distributed knowledge. Given a group of agents, their distributed group knowledge is the sum of all they know, including all bits of knowledge that only one agent knows. If agent a knows p and q, and agent b knows q and r, then their distributed knowledge is p, q and r. It is the knowledge every agent would have, if all agents would tell all other agents what they know.

Another kind of group knowledge is knowledge shared by every agent in the group. If agent a knows p and q, and agent b knows q and r, then everyone knows q.

The other kind of group knowledge is common knowledge. Common knowledge is knowledge which every agent in a group knows, but furthermore which every agent knows that every other agent knows that every other agent knows, ad infinitum.

In the framework of epistemic logic, given a group of agents G, let D_G denote an operator for distributed knowledge, E_G denote an operator for something everyone knows, and C_G denote an operator for what is commonly known as per the definitions above. I will define D_G and E_G by my own definitions, while C_G by [6], as I both find my own definitions of the two former as more intuitive and [6] as lacking the second part of both definitions.

Let $a \in G$. D_G can be semantically defined within epistemic logic in model $M = (S, R, V)$ as:
$M, s \vDash D_G\phi$ if for some agent a, $M, s \vDash K_a\phi$.
$M, s \vDash D_G(\phi_1 \bigwedge \phi_2)$ if $M, s \vDash D_G\phi_1$ and $M, s \vDash D_G\phi_2$.
Similarly, E_G can be semantically defined as:
$M, s \vDash E_G\phi$ if for all a, $M, s \vDash K_a\phi$.
$M, s \vDash E_G(\phi_1 \bigwedge \phi_2)$ if $M, s \vDash E_G\phi_1$ and $M, s \vDash E_G\phi_2$.

Let R_{E_G} denote the accessibility relation of E_G. This is simply the union of all accessibility relations for all agents. We will call the transitive closure over R_{E_G} for R_{C_G}.

[6] defines the semantics of C_G as the following:
$M, s \vDash C_G\phi$ if and only if for all t such that R_{C_G} st, $M, t \vDash \phi$

2.3 From knowledge to belief

This subsection is based on [7].

Knowledge is not the same thing as belief. To define a logical system for belief, we can use the same logical system as for knowledge, but we need to impose on it some concept of uncertainty.

That uncertainty can be measured in different ways. I will discuss briefly some different *measures* that can express this uncertainty. For more, see [7].

One measure we can use to express uncertainty, is the *probability measure*. We will go further through this concept more in detail later, but in short, a probability measure is a valuation from 0 to 1, grading the likeliness of an event to occur given a random sample. Probability gives a simple way of numerically represent uncertainty, and is well understood. In terms of epistemic logic, the weakness is that ignorance is not represented, and computing all probabilities may be costly.

Another measure we can use is *Dempster-Shafer belief functions*. Dempster-Shafer belief functions start with a set of subjectively chosen mass values, which sum to 1. Based on these, belief is calculated, given $W = all\ outcomes$ as

- $Bel(\emptyset) = 0$
- $Bel(W) = 1$
- $Bel(\bigcup_{i=1}^{n} U_i) \geq \Sigma_{i=1}^{n} \Sigma_{\{} I \subseteq \{1, ..., n\} : |I| = i\} (-1)^{i+1} Bel(\bigcap_{j \in I} U_j)$

Given the mass values we can also calculate a plausibility value of a set X as the sum of the mass of all intersecting sets of W. One benefit of belief functions is that they handle ignorance better than probability does.

Yet another way of representing uncertainty is the *possibility measure*. Possibility is similar to probability but differs in that while the union of two disjoint elements in probability is the sum of the the probability of each, the union of two disjoint sets (as well as joint sets) using a possibility measure is the maximum of the two:

- $Poss(\emptyset) = 0$
- $Poss(W) = 1$
- $Poss(U \bigcup V) = max(Poss(U), Poss(V))$, given U and V are disjoint.

An advantage of possibility measures is that they are less computationally demanding than probability.

Yet another way of representing uncertainty is *ordinal ranking functions*, which assign a value to all possible beliefs, and rank-order them from least believable to most believable, so that something that is believed with certainty has the value 1 and something that is believed to be impossible has the value 0. Define this as

- $K(\emptyset) = 0$
- $K(W) = 1$
- $K(U \bigcup V) = min(K(U), K(V))$, given U and V are disjoint.

Yet another possible measure is a non-numerical ranking order of *relative likelihood*.

Another possible measure is the *plausibility measure*, in which each world is assigned a value from 0 to 1 (inclusive) and partially ordered. This is similar to probability, but in addition there is a special value $\bot \leq 0$ and $\top \geq 1$.

As probability is well understood and intuitive to calculate with, I will be using a probability measure in this thesis. But it is important to recognize that there does exist flaws with it and that there also exists other alternatives.

2.4 Probability theory and Bayesian reasoning

Before going further, we should have a clear grasp of probability theory. This section contains an introduction to basic probability theory, see for instance [8].

Some psychologists [9] believe that humans have evolved in such a way that belief can be categorized into rough categories of probability, such as "impossible", "unlikely", "likely", "certainly". This is certainly the case for belief when discussed in everyday conversation, in which case belief sometimes can not be directly quantified by means of probability, although "certainly" and "impossible" sometimes fails. Nevertheless, as we are not as much interested in belief as a cognitive aspect of humans but as a computational device in multi-agent systems, we will be using numeric probability as the main tool for reasoning about belief.

When measuring something, we need to know three things: What is being measured, what part of it is being measured, and what are the specific measurements. In measure theory, a *measure space* consists of a triple, $(\Omega, \mathcal{F}, P)$.

Ω is the set of all possible outcomes for a random experiment. A variable X that is assigned value from a random experiment, is known as a *stochastic* variable.

$\mathcal{F}$ is a set of events which is a subset of Ω. An event is said to have occurred, if a stochastic variable is assigned a value in $\mathcal{F}$.

An *algebra* over Ω is a set $\mathcal{F}$, that includes the empty set, all atomic elements of Ω, and all possible combinations of union of said atomic elements.

For example, if the experiment is the flipping of 4 coins, and X denotes the outcome, where H denotes heads and T denotes tails, $\Omega = \{TTTT, TTTH, TTHT, TTHH, THTT, THTH, THHT, THHH, HTTT, HTTH, HTHT, HTHH, HHTT, HHTH, HHHT, HHHH\}$. An event "$X$ has at least 3 tails", corresponds to X being assigned a value within the subset $\{TTTT, TTTH, TTHT, THTT, HTTT\}$.

Finally, P is a mapping from events to a value, where P is a function $\mathcal{F} -> [0, \infty]$. $P(X)$ is a *measure* of event X.

A *probability space* is a measure space in which the sum of the measures of all possible outcomes is 1. That is, $P(\Omega) = 1$. In a probability space, we can more formally call $\mathcal{F}$ a *probability mass function*, or *pmf* for short.

In a probability space, the measure is called a *probability*, and denotes the relative frequency, inclusively between 0 and 1, that an event will happen in a stochastic variable for a random experiment. If for an event E, $P(E) = 0$, then the event E is impossible. If $P(E) = 1$, then the event is necessary. Any other value of $P(E)$ will give an increasing likeliness of event E occurring.

Within the theory of classical statistics, we can look at probability in terms of a stochastic model. Within this model, we often assume that each elementary outcome is equally probable. In this case, we have that

$$P(E) = \frac{|E|}{|\Omega|}.$$

In terms of the example with 4 coin flips, each outcome has an equal probability of 1/16, and $P(\text{"}X \text{ has at least 3 tails"}) = 5/16$.

When we do not have a formal stochastic model, we can often estimate the probabilities by relative frequencies of events when repeating an experiment a large number of times.

If event A and B are disjoint, then $P(A \cup B) = P(A) + P(B)$. Further,

- $P(\emptyset) = 0$,
- $P(\Omega \setminus A) = 1 - P(A)$.
- When A and B are not necessarily disjoint, $P(A \cup B) = P(A) + P(B) - P(A \cap B)$.
- $P(A \cap B) \leq P(A \cup B) \leq P(A) + P(B)$.
- If $A \subseteq B$, then $P(B \setminus A) = P(B) - P(A)$.
- If $\Omega = A_1 \cup \cdots \cup A_n$, then $P(B) = P(A_1 \cap B) + \cdots + P(A_n \cap B)$.

The *conditional* probability of A given B is defined as

$$P(A \mid B) \triangleq \begin{cases} \frac{P(A \cap B)}{P(B)} & \text{if } P(B) > 0, \\ 0 & \text{otherwise.} \end{cases}$$

The following multiplication rule:

$$P(A \cap B) = P(A)P(B \mid A)$$

is the basis of Bayes' law:

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

which is very useful for logic reasoning with probability-based knowledge, also called *inference*, which is the operation of inferring the value of a variable by observing another.

Let X be a *discrete*[1] stochastic variable that is assigned values from the set of outcomes $\Omega = \{x_1, \ldots, x_n\}$. This assignment is obviously easy if all outcomes are equally likely, as $P(x_i) = 1/n$ for i = 1,...,n. If they are not all equally likely, then a probability mass function $f_X : \Omega \to [0, 1]$ is defined by $f_X(x_i) = P(x_i)$, for each $i = 1, ..., n$.

We can form a joint stochastic variable XY, based on the two variables X and Y. X and Y are statistically independent, if and only if

$$f_{XY}(x_i, y_j) = f_X(x_i)f_Y(y_j)$$

Probability mass functions of joint variables may be expressed in the more general case by using the concept of conditional probability mass functions:

$$f_{X|Y}(x_i|y_j) \triangleq \frac{f_{XY}(x_i, y_j)}{f_Y(y_j)}$$

and hence,

$$f_{XY}(x_i, y_j) = f_Y(y_j)f_{X|Y}(x_i|y_j) = f_X(x_i)f_{Y|X}(y_j|x_i).$$

More generally, for a joint stochastic variable $X_1, \ldots, X_n$,

$$f_{X_1 \ldots X_n}(X_1, \ldots, X_n) = \prod_{j=1}^{n} f_{X_1 \ldots X_j}(X_j|X_1, \ldots, X_{j-1}) \tag{2.2}$$

Unfortunately, all terms in (2.2) can be exceedingly hard to compute if the variables $X_1, \ldots, X_n$ are not independent. This computation can however be simplified when some independencies between variables exist, as will be demonstrated in Section 2.7.

The *expectation* of an arbitrary function $F(X)$ is the sum of each outcome $\Omega = \{x_1, \ldots, x_m\}$ weighted by the probability, defined as:

$$\mathbb{E}[F(X)] = \overline{F(X)} \triangleq \sum_{i=1}^{m} F(x_i)f_X(x_i).$$

[1] In general, stochastic variables may also take values from a continuous range. For example, Ω may be a range of continuous real values. This means that a probability value will be formally zero for each possible value, and this makes it necessary to represent probability distributions in terms of probability densities, rather than probability mass functions. For simplicity and for the purpose of this thesis, the case of continuous variables is ignored.

The *variance* of X, in turn, represents the average distance from the mean, defined as

$$\mathrm{Var}(X) \triangleq \mathbb{E}\left[(X - \mathbb{E}[X])^2\right] = \sum_{i=1}^{m}(x_i - \mathbb{E}[X])^2 f_X(x_i) = \mathbb{E}\left[X^2\right] - (\mathbb{E}[X])^2.$$

2.5 Probabilistic Multi-agent Belief Logic

This section briefly discusses the logic of belief. For more, see [7].

A probabilistic multi-agent belief logic incorporates a *probability frame* into the epistemic logical framework. Formally, given a set of agents $\{Ag_1, ..., Ag_n\}$, a probability frame F has the structure $(W, Pr_1, ..., Pr_n)$ where W is the set of possible worlds and each Pr is a *probability assignment*. Pr_i is *equivalent* to the probability mass function $f_X(X = x_i)$[2]. A probability assignment in this context is a function which associates each possible world w with a probability space $\{W_w, F_w, \mu_w\}$, W_w being the world as seen from that state, F_w being the algebra over that world, and μ being the probability mapping.

To avoid confusion with the technical definition, I will give a quick example. Consider a situation where Alice and Bob are flipping a coin, and both Bob and Alice believe that the coin has an 80% chance of landing heads up. In this case, we have the possible worlds $W = \{heads, tails\}$, the algebra $F = \{\emptyset, heads, tails, \{heads, tails\}\}$, $Pr_{Alice} = Pr_{Bob} = \{W, F, \{\emptyset \Rightarrow 0, tails \Rightarrow 0.2, heads \Rightarrow 0.8, \{heads, tails\} \Rightarrow 1\}\}$. Here we let $\Rightarrow$ in the definition of the probability assignment denote the assignment of the left hand to the right hand probability value. Notice that each possible world here (heads and tails) have the same probability space, so the belief of the agents do not change from one to the other, but this needs not be the case.

We can make some constraints upon our probability assignments. Common constraints are:

$$\text{Uniformity: If } Pr_i(w) = (W_{w,i}, F_{w,i}, \mu_{w,i}) \text{ and v } \in W_{w,i}, \text{ then } Pr_i(v) = Pr_i(w)$$

$$\text{State-determined probability: If } v \in K_i(w) \text{ then } Pr_i(v) = Pr_i(w)$$

$$\text{Consistency: If } Pr_i(w) = (W_{w,i}, F_{w,i}, \mu_{w,i}), \text{ then } W_{w,i} \subseteq K_i(w)$$

We abuse K_i here to denote the operator from epistemic logic, so that it simple means "agent i knows".

[2]The two different notations will be used interchangeably for convenience and for consistency with two distinct strains of literature.

Uniformity is the notion that for each world agent i sees as having a probability of being the actual world, the probability space is the same.

State-determined probability is the notion that for any world an agent sees as possible the probability space is the same.

Consistency is the notion that only those worlds an agent sees as possible are assigned a probability.

2.6 Basic information theory

This section contains a brief introduction to information theory. A standard reference is [10].

Information theory grew out of the landmark paper [11] by Claude Shannon. The word "information" is used in everyday language in slightly different meanings. Shannon formalized it to signify "the amount of reduction of uncertainty about a variable obtained by observing the value of another variable."

Let X be a discrete stochastic variable with pmf (probability mass function) $f_X(x_j), j = 1, \ldots, m$ as defined in Section 2.4. "Uncertainty" here refers to the situation when the value of X is not yet known. Shannon used the word *entropy*, borrowed from thermodynamics, to formalize this concept of uncertainty. The entropy of X is[3] the statistical expectation of the negative of the logarithm of the probability of each outcome of X,

$$H(X) \triangleq \mathbb{E}[-\log X] = -\sum_{i=1}^{m} f_X(x_i) \log f_X(x_i) \tag{2.3}$$

By convention, the logarithm is assumed to be calculated in base 2 (that is, if $y = 2^z$, then $\log y = \log_2 y = z$), in which case the unit of measure of the entropy is a binary digit - a *bit*.

Example 1. *When $m = 2$ and the probabilities of the two outcomes are p and $1 - p$, respectively, the entropy function is conveniently expressed as a function of p. It is common to call this the* binary entropy function *$h(p)$, where*

$$h(p) = -p \log_2(p) - (1 - p) \log_2(1 - p).$$

The binary entropy function has a maximum value of 1 (bit) and is shown in Figure 2.1.

[3]Strange as it seems, this definition provides a useful measure of entropy which behaves consistent with intuitive "axioms" that one may impose.

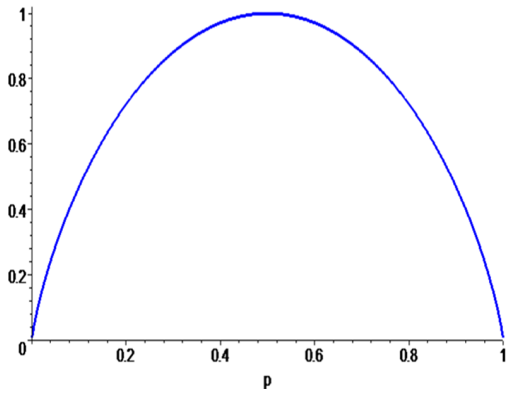

Figure 2.1: Binary entropy function $h(p)$.

Analogously, one can also define the conditional entropy $H(X|Y)$ of X with respect to some other variable Y,

$$H(X|Y) = -\sum_{i=1}^{m}\sum_{j=1}^{n} f_{XY}(x_i, y_j) \log f_{X|Y}(x_i|y_j),$$

where m and n, as before, are the number of possible values for X and Y, respectively. The *mutual information* between the two discrete random variables X and Y is defined as

$$I(X;Y) \triangleq \mathbb{E}[I(X = x; Y = y)] = \sum_{i=1}^{m}\sum_{j=1}^{n} f_{XY}(x_i, y_j) \log \frac{f_{X|Y}(x_i|y_j)}{f_X(x_i)} \tag{2.4}$$

It is easy to show that $I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$. The symmetry of this relation between X and Y accounts for the word "mutual". In words, the mutual information quantifies how much uncertainty about X (or Y, respectively) by observing the value of Y (or X, respectively.)

The significance of the mutual information $I(X;Y)$ with respect to inference is that it quantifies *how much* one can learn about X by observing Y. This quantification makes it possible to compare and select the most efficient among different methods of obtaining information.

2.7 Pearl's algorithm and belief propagation

Consider the competitive hat problem in Subsection 1.1.3 and its solution provided there. Many problems of distributing and interchanging knowledge among group members will take this form, although most practical problems will have a larger number of parameters

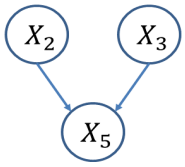

Figure 2.2: A DAG which is a tree.

involved. The brute force approach comes with an exponential complexity in the number of parameters, which is of course prohibitive for its application.

Thus there is a need for a more efficient algorithm to this type of problems. This section describes Pearl's algorithm [12, 13].

Recall from (2.2) that pmfs of are hard to compute in general, if elementary variables are not independent. If *some* variables are independent, however, the joint pmf can be factored as follows:

$$f_{X_1 \dots X_n}(X_1, \dots, X_n) = \prod_{j=1}^{n} f_{X_1 \dots X_j}(X_j | X_1, \dots, X_{j-1}) = \prod_{j=1}^{n} f_{\{X_j\} \cup parent(X_j)}(X_j | parent(X_j)). \tag{2.5}$$

Here, $parent(X_j)$ informally informally refers to "variables" on which X_j depends. More precisely, $parent(X_j)$ is the set of predecessors of X_j in the Bayesian network in Definition 2.

Definition 1. *A* directed acyclic graph *(DAG) is a graph with a finite number of nodes,* directed *edges, and no directed cycles (*i.e.,*, no directed paths from a node back to itself. If the DAG contains no cycles at all, it is a tree (see Figure 2.2). In general a DAG is not necessarily a tree (see Figure 2.3).*

Definition 2. *A Bayesian network is a DAG in which each node represents a unique stochastic variable, and each edge represents probabilistic dependency. Thus if there is an edge from node/variable Y to node/variable X in the Bayesian network, it means that $f_{XY}(X|Y)$ is a factor in the joint probability distribution. Implicitly, $f_{XY}(X|Y)$ labels this edge and must be a known conditional pmf, meaning that the calculation in (2.5) can be carried out if and only if the conditional probabilities $P(X = x|Y = y)$ must be known for all possible outcomes x for X and y for Y, and for all edges in the Bayesian network representing (2.5).*

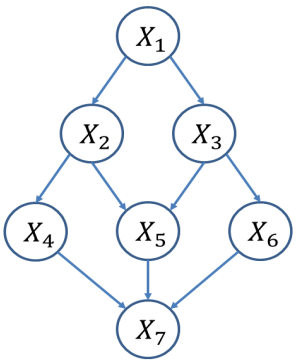

Figure 2.3: A DAG which is a not a tree.

J. Pearl's belief propagation algorithm [12, 13] works on a Bayesian network, which is defined above as a DAG that represents stochastic variables and the statistical dependence between pairs of them. The goal of the algorithm is to determine the *a posteriori* set of pmfs corresponding to the initial pmfs, using these known statistical relations. The description here is loosely based on [14].

Each node in the graph is associated with one stochastic variable X. Abusing notation, the node will for convenience be referred to as "node X". The node can perform calculations using and modifying a temporary estimate of the pmf $f_X()$, including versions of the pmf conditioned on other variables, as well as sending and receiving *messages* to and from parent nodes and child nodes. For example, node X_5 in Figure 2.3 will communicate directly only with the parent nodes X_2 and X_3, and with the child node X_7. Thus, the algorithm is a decentralized *message-passing* algorithm. In principle this means that agents in a group may act independently, cooperating according to the protocol imposed by the algorithm, but of course the algorithm may also be implemented on a single computer controlling all nodes.

Each node maintains a collection of local variables, including the pmf. For each adjacent edge, the node is assumed to know the conditional pmf $f_{X|Y}(x|y)$ for all neighbour nodes Y and for all possible values of x and y. In case a node has no parent nodes, the node will know its pmf.

When a node is activated, it "reads" the messages received from each of its parents and children, updates its belief based on these messages, and then sends new messages back to its parents and children. From its parent Y, a node X receives a message $\pi_{Y\to X}$ which essentially contains Y's current estimate of its pmf, that is, a list of probabilities, one for each possible outcome of Y. This estimate is based on the knowledge that has already reached Y by the action of the algorithm. From its child Y, node X receives a message $\lambda_{Z\to X}$ which essentially contains a list of nonnegative real numbers (likelihoods),

one for each possible outcome of X. This represents a distribution on X conditioned on what the child node Y knows so far.

Subsequently, node X updates its estimate of its own pmf using the input of all the neighbours. For each neighbour Y, node X also creates one version of its pmf based on the input from *all the other* neighbours of X, excluding Y, and sends this pmf to Y.

All nodes also keep track of some auxiliary variables which will be ignored for now, but which will be mentioned when they are relevant in the discussion in Chapter **??**.

At the termination of the algorithm, the current estimate of each node's pmf will be used as an estimate of X's pmf conditioned on all the other variables.

Algorithm 1 Pearl's belief propagation algorithm.

Require: Bayesian network representing stochastic variables and their dependencies
Ensure: Output estimate of pmf.

```
while More nodes to process do
  Select a node X that has received all messages from above, or all messages from below
  if the messages came from above then
    Calculate and send message π_{X→Z} for each child Z using equation (2.2)
  else
    Calculate and send message λ_{X→Y} for each parent Y using equation (2.2)
  end if
end while
Return current estimate f_{X|everyone else}() for each node X.
```

Kim and Pearl [12, 13] showed the following theorem:

Theorem 3. *If the Bayesian network is a* tree, i.e.*, does not contain any loops, and all variables have at most q possible values and at most m parents, then Algorithm 1 terminates with the correct probability mass function $f_{X_1 \dots X_n}(X_1, \dots, X_n)$ in $O(nq^e)$ operations.*

For a tree, the while-condition in the algorithm will become false after one pass in each direction through the tree. If the algorithm is applied to a non-tree, the while loop may be stopped by ad-hoc rules, for example if the results appear to be converging, or after a predetermined number of iterations. Note that a brute-force approach would require $O(q^n)$ operations. If the graph is *not* a tree, then the problem of computing the exact and correct pmf $f_{X_1 \dots X_n}(X_1, \dots, X_n)$ is known to be NP-hard. It has been observed empirically, however, that versions of the algorithm can perform well even if the restrictions that the graphs are loop-free DAGs are relaxed. For example, unless there are short cycles, the algorithm in practice performs excellently when applied to graphs used for decoding of error correcting codes (see for example [14]), even if those graphs

are undirected and contains cycles. These modern versions of the algorithm are designed to maximize the mutual information on edges, possibly modifying the graph through pre-processing first[14].

2.7.1 Pearl's algorithm applied to small examples

Pearl's algorithm applied to the hat problem in Section 1.1.3

Let x_i and y_i, $i = 1, \ldots, 3$ represent the hat colours and state of arm raising of player i. Player i wants to determine x_i but can observe only the other variables (including its own y_i which is irrelevant since it is computed from the other variables.) The problem can be solved through a crude invocation of Pearl's algorithm on the graph in Figure 2.4, where probabilities of the observable x_i and y_i are initially set to 0 or 1 according to direct observations. If player i reaches a conclusion, it will report to the *Game over* node which will inform the other players that the game is over. If the game has not ended in a reasonable time, players can claim a *red* hat.

Note that even this approach here is applied to the simple version of the problem, it may also work on noisy versions: For example, assume that observations may be incorrect due to foggy conditions, occurrences of the rather rare condition of red-blue daltonism, or cheating participants. Pearl's algorithm still provides a way to approach the correct result provided that the game's parameters are appropriately assessed.

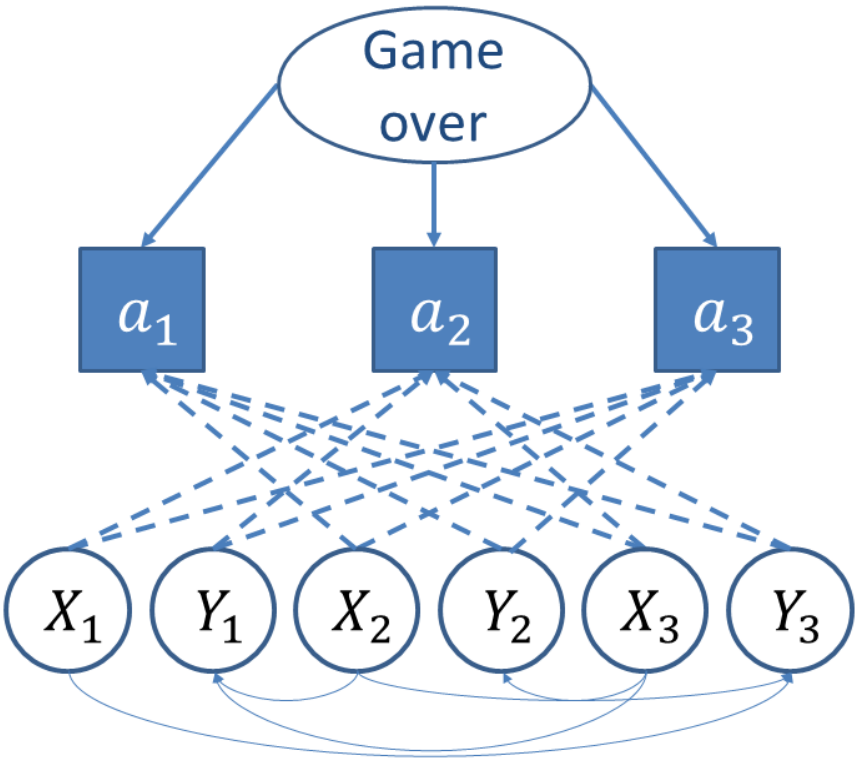

Figure 2.4: A Bayesian network for the hat problem in Section 1.1.3

Chapter 3

Defining belief in a practical sense

3.1 Acquisition of Belief

In our definition of Probabilistic Belief Logic, we have not mentioned where an agent may come about their probability. Before going further, we should dig into this subject as it is important to define in a practical sense what we are talking about when we talk about belief.

In a very loose sense, our definition of probabilistic belief is one which simply assigns a numerical value inclusively between 0 and 1 to an alternative state of the world.

One obvious way an agent may come about probabilistic belief is by statistical observation. For example, if the weather in a city seemingly arbitrarily changes from day to day so that it rains some days and is sunny other days, the agent may, in the most simple way, conclude that the chance that it will rain tomorrow is the amount of times that it has rained divided by the amount of days he has observed.

This would be an appropriate way to obtain probabilities at a lot of times, but not necessarily in a direct sense in all. For instance, let's say that Alice asks Bob out on a date. Alice has never before asked Bob out on a date. What can Bob say about whether or not Alice will be late for the date or not? Needless to say, Bob will not be able to deduce this information simply from statistical evidence on Alice being late for dates, apart from in the extreme circumstance where Alice has a vast history of being late for dates with other people of which information Bob is privy to.

This being said, even if Bob does not have any experience of going on dates with Alice, it still makes sense that Bob can have a belief of whether or not Alice would be late for a date. This is because he can draw belief not only from direct situational statistics,

but also from the trust he holds towards Alice. This is, in essence just a special case of statistical evidence. That is, Bob can conclude that given the statistical evidence of the things Alice has said that (seemingly) was true, this new thing that Alice said should have a likeliness of $T = C/A$, where T is a trust value for the agent Alice, C =(all the correct things Alice have said), and A =(all the things Alice have said).

In an analytical sense, it would then make sense to store and update such a T-value for each agent in the system in which an agent operates.

As the acquisition of agent-specific belief probabilities is not the subject of this thesis, I will not go much further into this subject than this, but will use the following definition as the basis for what is meant by acquired probabilistic belief in this thesis:

Definition 1. *Let an agent's* trust *in another agent denote a value between (inclusive) 0 and 1, where 0 is a total disbelief and 1 is a total belief in that the other agent will act in accordance with the agreed upon or socially optimal outcome.*

Definition 2. *Let an agent's* outcome belief *denote a value inclusively between 0 and 1 acquired either through the result of an experiment relating to the probabilistic likeliness of an event.*

Definition 3. Belief *is the union of the set of an agents trust and its outcome belief.*

While belief is defined to be arguably two different things, I will assume that the two will not occur simultaneously and also act in the same way when an agent is deducing what to do. That is, I assume that if our agent trusts an agent to do x with a 0.5 likeliness, this is essentially the same as having observed that the agent does x 50% of the time.

3.2 Multi-Agent Belief Logic

While we in the background chapters we have defined belief, we have not defined the syntax nor semantics of a logical language. We will try to do that now. Let M be a model $(S, Pr_1, ..., Pr_n, V)$, $ag \in Ag_1, ..., Ag_n$ an agent corresponding to one of the $Pr_1, ..., Pr_n$, and V be a valuation $V(s)(a)$, where $a \in At$ as described in 2.2.2.

To start with we will say that, with the first part being identical as in 2.2.2 from [6], probabilistic multi-agent belief logic can be defined (loosely) as:

- $M, s \vDash p$ if and only if $V(s)(p)$ is true.
- $M, s \vDash \neg p$ if and only if not $M, s \vDash$ p.

- $M, s \vDash p \bigwedge q$ if and only if both $M, s \vDash p$ and $M, s \vDash q$.
- $M, s \vDash B_{ag_n,x}(\phi)$ if $(\phi \Rightarrow y) \in \mu_{ag_n}$ so that $y \geq x$.
- $M, s \vDash B_{ag_n,x}(B_{ag_m,y}(\phi))$ if agent n believes with a certainty of greater than or equal to x that agent m believes with a certainty of greater than or equal to y that ϕ.

3.3 Common Belief

Remember that common knowledge is not defined as being knowledge held by everyone in a group; it is defined as knowledge held by everyone within a group, which everyone in the group knows that everyone in the group holds, and which everyone in the group knows that everyone in the group knows that everyone in the group holds, ad infinitum.

How should we then define common belief? The intuitive notion would be to define it exactly as with common knowledge, but exchanging the 100% relationship with a a% relationship. That is: Everyone believes x with a b% likeliness, and everyone believes that everyone believes x with a b% likeliness, and everyone believes that everyone believes that everyone believes x with a b% likeliness, ad infinitum. Here, every belief relation has a likeliness of at least a%.

Notice that we have defined the belief operator in a very loose way. But this definition makes something clear. Going about defining the operations of belief in the same way as in epistemic logic, we run into a problem. Namely, what can we say about the belief a given agent has about other agents' beliefs? To say anything about an agents' belief about another agents' belief, it would seem like we would need to keep a database of knowledge about all agents' beliefs, and all agents' beliefs about all agents' belief, and all agents' beliefs about all agents' belief about all agents' beliefs, and so on. This was somewhat straightforward in epistemic logic when the probabilities of each step is the same (1), but not when the probabilities of each step is different.

As an example, take the Two Army Problem. Say that at some point in the message swapping back and forth, each general believes with a belief of 0.3 that the opposite general believes with a belief of 0.3 that the opposite general, ..., believes with a belief of 0.3 that the opposite general knows to attack at the specified time. If we were to keep all of these values stored in some formal representation of the problem using the same approach as when defining common knowledge in epistemic logic, we would at least need to store the probability assignment for each agents belief. But what about the belief about an agents belief? Since the belief value can be any value between 0 and 1, this value can take an infinite amount of values. You could cast this onto a discrete space by partitioning the infinite values between 0 and 1 into a discrete set such as

$\{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$. But for each next step, this blows up even further, even if the probability space is made so that it is not infinite.

3.4 Defining Common Belief

The way I propose that common belief should be defined, is as a single inclusive number from 0 to 1, representing how much all agents believe other agents believe something.

Proposition 1. *The strength of all agents' common belief of ϕ, is the number greater than or equal to 0 and smaller or equal to 1, describing the mean likeliness that all agents believe ϕ.*

3.5 The Strength of Another Agents Belief about Another Agents Belief

If each level of common belief is partitioned into n different segments of the same size of probability values, reasoning about the first level of m other agents, belief would require the explicit memorization of nm probability distributions. The second level of common belief would then need nm^2 probability distributions. On the i'th level of common belief, we have nm^i probability distributions. Common belief in its perfect form would need an infinite such levels, which would require the explicit memorization or calculation of nm^∞ probability distributions.

This leads us to the following claim:

Claim 1. *An agent's belief about another agents belief can not be formalized recursively. An agent needs to explicitly calculate and remember each level of belief.*

This suggests that somehow short-circuiting the probabilistic reasoning based on at most a few levels of explicit beliefs is the only way to go about handling common belief since explicitly calculating it is mathematically untenable. Two possible such short-circuits jumps out as likely. Firstly, somewhat unlikely, is the possible that the probability, or strength of the belief, goes towards 0 or 1 so that one at some point do not need to calculate further. The second, perhaps more likely is that the strength of the belief goes towards some limit value, in the case above that of 0.3, so that one at some point do not need to calculate further.

Definition 4. *Let a* belief level*, or level for short, denote the depth of belief about other agent's belief, that is, -1 plus the number of B's involved in the belief logical statement so that $B(B(...(\phi)...))$.*

Definition 5. *Let* belief depth*, or depth for short, denote the amount of levels the main agent has memorized.*

Definition 6. *Let AB be the set of probability spaces for each agent's belief for the first level of belief. Let* AB_{ag}*, given* $ag \in agents$*, be the probability distribution defining belief for a specific agent. Let n be a fixed number of divisions of belief, within the probability range 0 to 1 so that each part has the size* $\frac{1}{n}$*. Then, let* $AB1$ *be a set of distributions over the set* $\{0, \frac{1}{n}, ..., \frac{n}{n}\}$ *(notice that since this is a probability distribution, the sum of the probability of each chunk sums to 1), one for for each distribution in AB for each other agent including the main agent. In the case of* $AB1$*, a single case is subscripted with one more agent than AB, so that* $AB1_{ag1,ag2}$ *is the belief about* $ag1$*'s belief about* $ag2$*. Define ABi to be the i'th level, defined in the same way as* $AB1$*. A single case needs to be subscripted with one more agent than the previous level of AB.*

3.6 Combining the Probabilities of Beliefs and Nested Beliefs

If we assume all agents are rational and honest, then we could multiply our own belief about ϕ with all other agents' belief, and divide it by the number of agents. The result would be a mean of the beliefs.

Similarly we could make a calculation over $AB1$, averaging over all products of the highest probability values of $AB1_{ag1}ag2$ and the probabilities from AB to get the mean of what all agent's believe about what all agents believe. Then for the next level we can make the same calculations, down to our depths, to get a measure of common belief.

The problem with this, is that that it does not account for conditional dependencies. So instead, let us use a slightly different approach.

3.7 Bayesian Networks as Common Belief

An alternative way to view the first level of common belief, would be as a bayesian network, in which a node for each agent is represented as a dependency for an agents common belief.

For example, let us say we have three agents, of which one is the one we are reasoning for, call it a. Call the other two agents $a1$ and $a2$. Call what is being believed x. Then we could think of the belief as the following bayesian network that describes the situation:

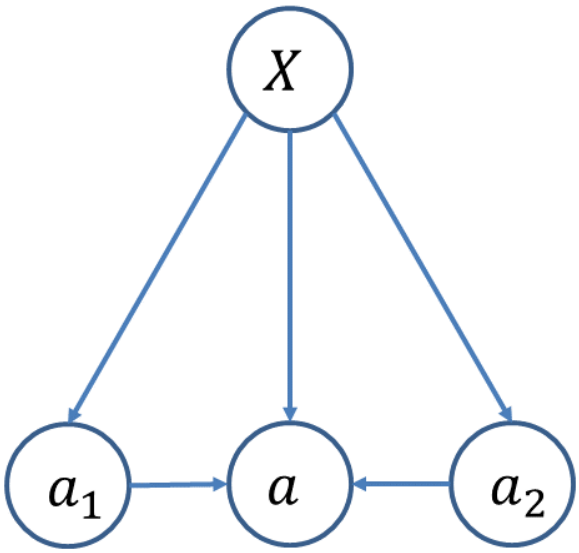

Figure 3.1: A Bayesian network for the first level of common belief

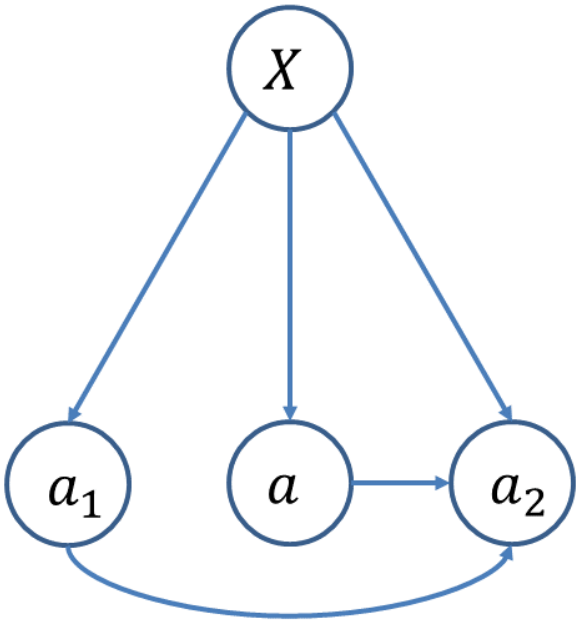

Figure 3.2: One bayesian network for the second level of common belief

A bayesian network shows us dependencies. Notice that we are abusing the terminology a bit here; a here is our common belief. The common belief depends on x, or rather agent a's belief about x, of agent a's belief about $a1$'s belief, and of agent a's belief about $a2$'s belief.

Then what about the next level? The next level would then have to be represented similarly by one similar bayesian network each for $a1$ and $a2$, which are depicted in Figure 3.2 and Figure 3.3.

The next level can further be derived from the second levels, and so on. We can see that the complexity clearly would explode for each new level. Any agent reasoning explicitly about common belief would have to limit the reasoning at some depth. Because of the complexity increase each level, the depth at which the reasoning is limited must also be at some shallow depth.

Claim 2. *Because of complexity issues, in practice common belief must be dealt with using belief with a shallow depth, optimally with the lowest level being AB*1.

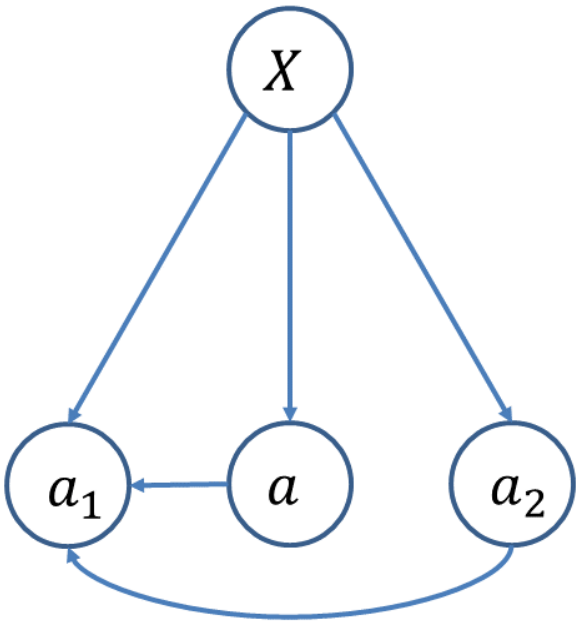

Figure 3.3: The other bayesian network for the second level of common belief

3.8 Acquisition of Common Belief

How does an agent come about common belief? One obvious way, would be that the belief is proclaimed in such a way that an agent can assume that all other agents in a group have heard it and are aware that the other members of the group also heard it. A special case of this would be the meta information of rules given a game example, where everyone knows that everyone else knows the rules.

But this is a very limited way of acquisitioning common belief, as some information may not be proclaimed. It makes sense to assume that some, if not most, acquisition of common belief is acquired through introspection of what other agents may believe.

3.9 Analyzing Common Belief through Pearl's Algorithm

When we think of group belief in terms of a bayesian network, we could potentially use Pearl's algorithm to come about a strength of common belief. There is one initial, possibly big problem with this. Pearl's algorithm is intended for use on a directed acyclic bayesian network. Therefore it may give the wrong results when applied to a cyclic graph, such as the graph for common belief must be. Keeping this in mind, let us move forward.

The graph of the bayesian network for common belief must be cyclic, since combining all the levels and each level of belief into one, we get a bayesian network in which all nodes are connected to all nodes except what is known. Figure 3.4 depicts such a bayesian network given two other agents and a given belief X. Other probabilistic dependencies could also be added.

Notice that 3.4 only uses belief down to $AB1$. While this may lead to inaccuracies, this way of handling common belief is far more efficient than not making this limitation, as it holds true to 2.

Since we are dealing with a directed cyclic graph, the algorithm needs to be tweaked to accomodate that. The algorithm in Algorithm 2 is an attempt at making an algorithm for a cyclic graph.

This algorithm calculates an averaged probability mass function based only on AB and $AB1$, and not on further levels of ABi such as $AB2$ and so on. The resulting probability mass function represents the strength of common belief.

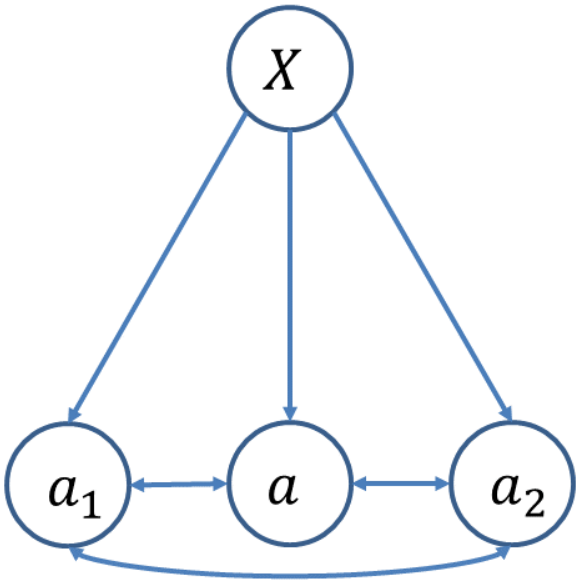

Figure 3.4: A Bayesian network for common belief

Algorithm 2 Pearl's belief propagation algorithm for common belief.

Require: Cyclic bayesian network representing stochastic variables and their dependencies.

Ensure: Output estimate of pmf.

while More nodes to process **do**

Select a node X that has received all messages.

Calculate the mean of all probability distributions and send to all others.

if $f_{X|\text{everyone else}}()$ has not changed since last iteration OR some time limit t is overstepped **then**

Return sum of current estimate $f_{X|\text{everyone else}}()$ for each node X, divided by number of agents.

end if

Update estimate of $f_{X|\text{everyone else}}()$

end while

3.10 Why limit the calculation of the strength of common belief to $AB1$?

In certain cases beside common knowledge, it makes sense to talk about levels of knowledge deeper than $AB1$. For example, we may want to describe an agent who believes that agent two believes that agent three believes that it is raining. Because of that, even if we do not use those levels in calculations regarding common belief, it still makes sense for the agent to remember and reason about knowledge deeper than $AB1$. Then why would we limit our calculation of the strength of common knowledge to a bayesian network of $AB1$?

The reason I have chosen to do this is that while each layer more finely grains the beliefs, it also increases the complexity of the calculations by a factor of —*agents*— per level you go. Any change in the beliefs at any point in the resulting tree would call for a recalculation of the common belief, which would be immensely computationally costly. My assumption is that the resulting strength would not change by much, even if you perform a calculation including levels past $AB1$.

3.11 Practical Application of the Algorithm

Let a set of agents be $\{a_1, a_2, a_3\}$, and let a_1 be the agent performing the reasoning. Let X be a single bit stochastic variable of information, with the possible values $\{0, 1\}$. Let $AB0 = \{\{0.1, 0.9\}, \{0.2, 0.8\}, \{0.3, 0.7\}\}$ be the first level of agent belief a_1 has, so that the i'th element is the belief a_i has for X. Let $AB1 = \{\{\{0.1, 0.9\}, \{0.2, 0.8\}, \{0.3, 0.7\}\}, \{\{0.3, 0.7\}, \{0.2, 0.8\}, \{0.3, 0.7\}\}, \{\{0.3, 0.7\}, \{0.3, 0.7\}, \{0.3, 0.7\}\}\}$, where $AB1_{i,j}$ corresponds to agent i's belief about agent j's belief. Notice that $AB1_{i,i}$ is simply the same as AB_i:

Proposition 2. *$AB1_{i,i}$ is the same as AB_i. Because of this, we do not need to know $AB0$ for our approach to common belief, only $AB1$.*

This situation is the following: Agent 1 believes that X is 90% likely to be true. Agent 1 believes that Agent 2 believes it to be true with 80% likeliness. Agent 1 believes that Agent 3 believes X to be true with 70% likeliness. For all unknown values, Agent 1 assigns a 70% belief that X is true. This gives us the bayesian network in Figure 3.5. x1, x2 and x3 constitute $AB0$ while the rest constitute the rest of $AB1$.

In relation to this depiction, $AB1 = \{\{x1, x6, x9\}, \{x4, x2, x5\}, \{x7, x8, x3\}\}$.

The algorithm then works like this: Choose an arbitrary node, say $A1$. Calculate the mean of probability distributions from others: For $A1$, this is $(x1 + x6 + x9)/3 =$

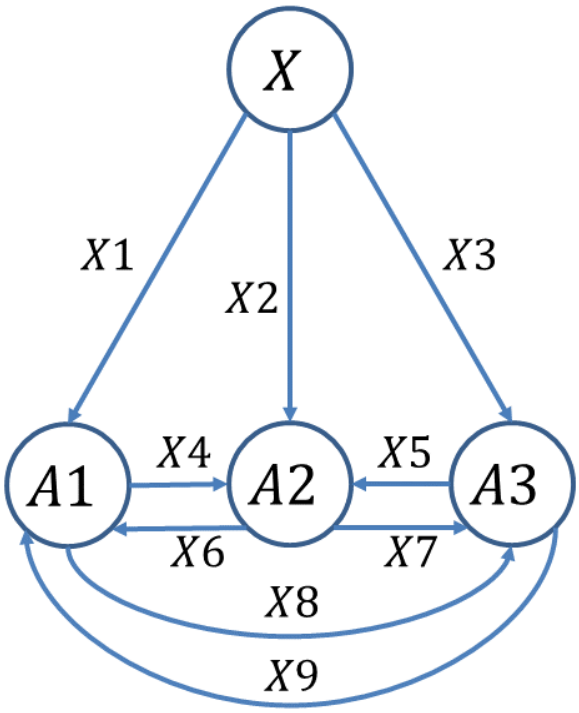

Figure 3.5: The bayesian network for our example

$\{0.6, 2.4\}/3 = \{0.2, 0.8\}$. Replace $x4$ and $x8$ with this new value.

Next, for $A2$, sum $\{x4, x2, x5\}$. This adds up to $\{\{0.2, 0.8\} + \{0.2, 0.8\} + \{0.3, 0.7\}\} = \{0.7, 2.3\}$ which divided by 3 is $\{0.23, 0.77\}$. Send this as the new value for $x6$ and $x7$.

Then for $A3$, $x7 + x8 + x3 = \{0.23, 0.77\} + \{0.2, 0.8\} + \{0.3, 0.7\} = \{0.73, 2.27\}$ which divided by 3 is $\{0.24, 0.76\}$. Send this again as the new values for $x5$ and $x9$.

After our first iteration we now have the new values:
$x4$ and $x8$: $\{0.2, 0.8\}$
$x6$ and $x7$: $\{0.23, 0.77\}$
$x5$ and $x9$: $\{0.24, 0.76\}$

Now return to $A1$ and repeat the calculations, so that $x1 + x6 + x9 = \{0.1, 0.9\} + \{0.23, 0.77\} + \{0.24, 0.76\} = \{0.57, 2.43\}$ which divided by 3 gives $\{0.19, 0.81\}$. Send this as the new messages $x4$ and $x8$.

Next, for $A2$ we have $x4 + x2 + x5 = \{0.19, 0.81\} + \{0.2, 0.8\} + \{0.24, 0.76\}$ divided by 3 which gives us $\{0.21, 0.79\}$. Send this as the new message to $x5$ and $x9$.

Next, for $A3$ we have $x7 + x8 + x3 = \{0.21, 0.79\} + \{0.19, 0.81\} + \{0.3, 0.7\}$ divided by 3 which gives us $\{0.23, 0.77\}$. Send this as the new message to $x5$ and $x9$.

After our second iteration we now have the new values:
$x4$ and $x8$: $\{0.19, 0.81\}$
$x6$ and $x7$: $\{0.21, 0.79\}$
$x5$ and $x9$: $\{0.23, 0.77\}$

Again return to $A1$ and repeat the calculations, so that $x1 + x6 + x9 = \{0.1, 0.9\} + \{0.21, 0.79\} + \{0.23, 0.77\} = \{0.54, 2.46\}$ which divided by 3 gives $\{0.18, 0.82\}$. Again, set this as the new messages $x4$ and $x8$.

For $A2$ we have $x4 + x2 + x5 = \{0.18, 0.82\} + \{0.2, 0.8\} + \{0.23, 0.77\}$ divided by 3 which gives us $\{0.2, 0.8\}$. Send this as the new message to $x6$ and $x7$.

Next, for $A3$ we have $x7 + x8 + x3 = \{0.2, 0.8\} + \{0.18, 0.82\} + \{0.3, 0.7\}$ divided by 3 which gives us $\{0.23, 0.77\}$. Send this as the new (unchanged) message to $x5$ and $x9$.

After our third iteration we now have the new values:
$x4$ and $x8$: $\{0.18, 0.82\}$
$x6$ and $x7$: $\{0.2, 0.8\}$
$x5$ and $x9$: $\{0.23, 0.77\}$

Again return to $A1$ to repeat the calculations, so that $x1 + x6 + x9 = \{0.1, 0.9\} + \{0.2, 0.8\} + \{0.23, 0.77\} = \{0.53, 2.47\}$ which divided by 3 gives $\{0.18, 0.82\}$. Again, set this unchanged value as the new messages $x4$ and $x8$.

For $A2$ we again have $x4 + x2 + x5 = \{0.18, 0.82\} + \{0.2, 0.8\} + \{0.23, 0.77\}$ divided by 3 which gives us $\{0.2, 0.8\}$. Send this unchanged value as the new message to $x6$ and $x7$.

Next, for $A3$ we have $x7 + x8 + x3 = \{0.2, 0.8\} + \{0.18, 0.82\} + \{0.3, 0.7\}$ divided by 3 which gives us $\{0.23, 0.77\}$. Send this as the new (unchanged) message to $x5$ and $x9$.

After our fourth iteration we now have the new values:
$x4$ and $x8$: $\{0.18, 0.82\}$
$x6$ and $x7$: $\{0.2, 0.8\}$
$x5$ and $x9$: $\{0.23, 0.77\}$

Our values has now converged at these values, so we add together each pmf, and divide by number:

$x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 = \{0.1, 0.9\} + \{0.2, 0.8\} +$
$\{0.3, 0.7\} + \{0.18, 0.82\} + \{0.23, 0.77\} + \{0.2, 0.8\} +$
$\{0.2, 0.8\} + \{0.18, 0.82\} + \{0.23, 0.77\} = \{1.82, 7.18\}$

$\{1.82, 7.18\}/9 = \{0.2, 0.8\}$

As a result of the analysis, we have that the (strength of the) common belief of X being the case, is 0.8.

3.12 Overview

This section is intended as an overview of what has been discussed in this chapter.

To summarize, when talking about beliefs about beliefs we can divide such beliefs into different levels. I use the terminology AB to denote the level, where $AB0$ is the beliefs the main agent has about all agents' beliefs. $AB1$ denotes the beliefs which require a two agent path to define, while $AB2$ denotes beliefs which require a three agent path to define, and so on.

While explicit beliefs should deal with deep levels of AB, I postulate that you ONLY need $AB1$ in the calculation of common belief strength (where AB is included in $AB1$).

If looking simply on $AB1$, you can formulate the situation as a bayesian network, albeit a cyclic one. We can then perform a modified version of Pearl's algorithm for belief propagation, to calculate a good approximation of the strength of common belief.

Chapter 4

Out in the Wild: How to expand my work

4.1 Dependencies in the extended Pearl's algorithm for cyclic graphs

Not much thought has been put in as to how and what dependencies for belief I should use in the algorithm. The algorithm should be expanded by formalizing this.

4.2 Complexity

While the complexity of our analysis is quite low due to our insistence of limiting the depth involved in analyzing common belief to $AB1$, one aspect can be possibly be improved by going past $AB1$. By going past $AB1$, we can possibly improve the accuracy of the calculation of the strength of common belief, but likely to no real benefit.

Chapter 5

Wild thoughts

5.1 $AB1$ in Relation to Human Common Belief

While I used $AB1$ mainly as a mechanism for keeping down complexity in calculating a degree of common belief, it is still an interesting proposition whether or not this may be the way human cognition treats common belief. People would almost certainly have a limited capacity of going to deep levels of ABi to reason about common belief, but certainly are able as groups of recognizing what a group believes as possible.

One thing I did not account for in my practical analysis of the modified Pearl's algorithm is the updating of the probability distributions for $AB0$. In our use, that was fine, but $AB0$ would certainly change in practice. An interesting context for this, would be the rise of social media, in which people will be showered with opinions they agree with, both because they choose their own friends and because the social medias also attempt to give the users what they are likely to want. If we see the opinions visible on social media as $AB0$, social media and the users all attempt to see opinions the users agree with. Then this in effect may be seen as the values of $AB0$ going towards the belief distribution of the user. This may polarize the beliefs of groups, leading to factions of people with a belief of what will happen and what should happen.

5.2 Will the algorithm give the correct result given a bayesian network that is not a tree?

One particular concern with the algorithm is that Pearl's algorithm is originally not intended for use on a cyclic graph, but only on a tree. Will it give the correct result

if used on a cyclic graph? In the examples tested the algorithm has converged at a meaningful number, but it may have flaws that I have not noticed that may occur in other circumstances. Research remains to be done as to whether the algorithm will actually work on a cyclic graph, but so far everything points in the direction that it does.

Chapter 6

Conclusion and further ideas

In conclusion, probabilistic belief in a multi-agent system can be given a degree, describing how much all agents in the system believe that all other agents believe all other agents believe, ad infinitum, that some proposed truth is the actual truth. In epistemic logic, such common belief is defined inductively into the infinite, but since we with probabilistic multi-agent belief are dealing with a multitude of different probability distributions, every additional level we take into consideration when calculating a numerical probability distribution for common belief multiplies the complexity by at least a factor of the number of agents.

Instead, this degree can be calculated by using our version of Pearl's algorithm for belief propagation on a cyclic bayesian network representing $AB1$. This certainly removes some of the accuracy of the calculation of the probabilities, but is an inexpensive way to calculate a the value that would otherwise be very expensive to calculate, with a result which approximates the real value.

Bibliography

[1] Martin J. Osborne. *An Introduction to Game Theory*. Oxford University Press, 2004.

[2] *Prisoner's dilemma*. Accessed at May 25, 2019. URL: https://en.wikipedia.org/wiki/Prisoner%5C%27s_dilemma.

[3] *Two Generals' Problem*. Accessed at May 25, 2019. URL: https://en.wikipedia.org/wiki/Two_Generals%5C%27_Problem.

[4] *Hat puzzle*. Accessed at May 24, 2019. URL: https://en.wikipedia.org/wiki/Hat_puzzle.

[5] L.T.F. Gamut. *Logic, Language and Meaning: Volume I, Introduction to Logic*. The University of Chicago Press, 1991.

[6] Hans van Ditmarsch et al. *Handbook of Epistemic Logic*. College Publications, 2015.

[7] Joseph Y. Halpern. *Reasoning About Uncertainty*. The MIT Press, 2003.

[8] Arnljot Høyland. *Sannsynlighetsregning og statistisk metodelære*. Tapir, 1979.

[9] Daniel Kahneman. *Thinking, fast and slow*. Farrar, Straus and Giroux, 2011.

[10] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory 2nd Edition*. Wiley, 2006.

[11] Claude E. Shannon. "A Mathematical Theory of Communication." In: *Bell System Technical Journal* (1948), pp. 190–193.

[12] J. H. Kim and J. Pearl. "A computational model for causal and diagnostic reasoning in inference systems." In: *IJCAI* (1983).

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

[14] R. J. McEliece, D. J. C. MacKay, and J.-F. Cheng. "Turbo decoding as an instance of Pearl's 'belief propagation' algorithm." In: *IEEE Journal of Selected Areas in Communication* 16 (Feb. 1998), pp. 140–152.