MONASH University

Interest-Relative Induction

Gerhardus Visser

Bachelor of Computer Science (University of Melbourne)

Honours degree of Bachelor of Computer Science (Monash University)

A thesis submitted for the degree of Doctor of Philosophy at

Monash University in 2016

Faculty of Information Technology

Copyright Notice

Abstract

Inference can be summarised as the act of deriving a conclusion from some premise. Induction is a class of inference which has many interpretations. Inductive inference goes beyond what can be deduced with certainty from within a given system of logic; it creates conclusions which generalise, explain or predict. This thesis presents an interest-relative account of inductive inference.

For inference to have a realistic context, there must be an agent which performs the inference and an agent which uses the conclusion produced. The term 'interest' is used here to refer to the practical investment and limitations of these agents. Practical investment is what the agents want, and practical limitations are what they are and are not capable of.

It is argued that most applications of induction, with the exception of pure prediction, are interest-relative. Those properties that are considered essential to common classes of induction – including explanation – can only be defended when the practical limitations applying to these agents are considered and when they are assumed to have limited investment. Limited investment here means that the agents are concerned with applying conclusions to specific environments.

While the idea that knowledge or belief can be interest-relative is present in epistemology, philosophy of mind and philosophy of science, it is currently not adequately explored in a way that is compatible with, and easily applicable to, fields directly concerned with designing inference algorithms. Discussions touching on this topic currently rely too much on the use of vague analogies or explanations through very specific examples; which leads to conclusions that lack rigour and generality.

A comprehensive analysis of the role of interest in inductive inference is needed. The concepts needed for an understanding of this topic need to be identified and developed. A solid interest-relative foundation for inductive reasoning must be developed.

A method for analysing the relation between induction and interest is introduced. Our method works by defining a formal model of interest and how it relates to inductive inference. The model employs imprecise Bayesian probabilities and decision theory. It has been refined over time to be as general and simple as possible while still capturing the essential relation. We demonstrate how this approach can be used to elicit a deeper and more precise understanding of that relation.

Based on what is learned from this analysis, the discussion moves to select topics to show how understanding and acknowledging the role of interest can aid the process of developing inductive inference algorithms. Specific areas that are looked at include: the concept of Occam's razor, the concepts of explanation and prediction, application of the minimum message length principle, and the minimum Kullback-Leibler divergence estimator.

An approach to the design of inductive inference algorithms which better acknowledges the role of interest is advocated. We advocated that more explicit efforts be made – when designing and defending inference algorithms – to describe the interest that applies and to relate design choices to it. This is demonstrated using decision trees.

Declaration

This thesis contains no material which has been accepted for the award of any other degree or diploma at any university or equivalent institution and that, to the best of my knowledge and belief, this thesis contains no material previously published or written by another person, except where due reference is made in the text of the thesis.

Acknowledgements

I would like to express gratitude towards my supervisor David Dowe who always endeavoured to put me in touch with other researchers, and to direct me to research and work opportunities. I thank my co-supervisor Toby Handfield who listened to my ideas and often pointed me in helpful directions that I would not have considered on my own. I also thank Pat Dale who helped during my early attempts to get involved in practical data analysis. I am grateful to my parents Rita Visser and Cobus Visser who were supportive at times when the project became difficult. I also thank John Buruma who took on the very large task of proofreading the thesis.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Intuitive Examples

The term "interest-relative" is one borrowed from epistemology. An interest-relative account of knowledge was defended by Stanley [2005]. There, the concept is defined as: "*bare interest-relative invariantism (henceforth IRI) is simply the claim that whether or not someone knows that p may be determined in part by practical facts about the subject's environment.*" Note that our work is not concerned with knowledge as used in epistemology, but with what the best belief to infer to is. While the basic premise in our work is similar, the method of analysis and the goals are very different. This rest of this section gives informal examples to help the reader form some intuition about these concepts. More general definitions will be developed after the thesis overview (section 1.5). For now, think of our goal as placing inference in a realistic context by considering who (or what) makes the inference and who (or what) uses its conclusion; as well as considering how they communicate.

The first example is loosely based on an example from [Stanley, 2005]. Imagine that internet banking doesn't exist yet and that on Monday, both Sarah and Tim ask you if the local bank branch will be open on Friday. You may answer either yes or no, they will believe your answer without doubts and act as if your estimate is certain. This 'yes-or-no' requirement might seem unreasonable but the reasons for it will be elaborated on in the following paragraphs. You feel there is roughly a 95% chance that it will be open

but you may only answer yes or no. You know that Tim has a bill to pay and if the bank is closed on Friday he can try again some other time without significant repercussions. You know that Sarah has a bill to pay and that she may be evicted from her home if it is not paid before Saturday. You answer yes to Tim and no to Sarah.

Both answers are based on the same observations and prior beliefs (basic assumptions about the workings of banks). Both Tim and Sarah will act as if your answer is true exactly. The set of possible answers, yes or no, include all possibilities; exactly one of them is true. The best answer – from the allowed set of answers – differs for the two individuals only because they have different investment in the accuracy of the prediction.

The problem here occurs because the language of allowed answers, in this case yes or no, is not expressive enough to capture the degrees of uncertainty or special cases which might be relevant to both Tim and Sarah.

If the language of allowed answers is made more expressive, there may exist a single answer that is appropriate for both Tim and Sarah. Imagine that the language is extended to contain: highly probable, probable, fifty-fifty, improbable, and highly improbable. Now, the answer "probable" may be the best estimate for both Tim and Sarah. Another way to adjust this language is to include "I don't know" as an acceptable answer. It can be seen that, for this example, broadening the language of allowed answers reduces the interest-relativity of the best answer.

Imagine that you have called the bank to ask if they will be open on Friday and, given their response, you update your belief to roughly 99.9% probability that it will be open. Now the answer "yes" may be appropriate for both Tim and Sarah. In this example, the interest-relativity, caused by a restrictive language of answers, is reduced as more information becomes available.

These examples beg a question: For a given a data source, will there always be a language of answers that removes all interest-relativity? Our answer is yes; but, using such a language will tend to conflict with the goal of making inferences that count as explanations (see section 3.1). Note, the bank example is concerned with prediction and not explanation.

Realistically, when asked if the bank will be open on Friday, one would probably answer in more detail even if the question was clearly stated as a yes-or-no question.

The level of detail one gives when replying to such questions depends on context. There are, however, cases where the level of detail really is restricted. A sportsman indicating his intentions to a team-mate does not have time to elaborate and may be restricted to a small set of gestures. A researcher asked to write a one paragraph summary of his thesis will almost certainly end up either misleading someone or saying too little to be satisfactory. When answering a multiple choice question on a survey, there is no option to elaborate. When designing automated systems – as used in artificial intelligence, statistics and machine learning – the forms of inferred conclusions must often be limited to fit some narrow predetermined schema.

Returning to the bank example, one might answer very elaborately; if the temperature goes over 50 degrees Celsius there is a 20% chance that it will be closed; if there is a flood it is 10%, actually, it depends on the severity of the flood .... Knowing when and how to elaborate is intuitively easy in every day communication about familiar topics. We know when and how to elaborate when the goals and limitations of the person being communicated to is understood. Capturing this ability formally such that it may be programmed is difficult.

Relating this problem to everyday communication is further complicated by the fact that sentences communicated between individuals are seldom interpreted entirely literally. When telling Sarah that the bank will be open, she knows that you cannot be absolutely certain and takes that into consideration. With an automated system, such intuitions might not be present. Furthermore, it is unclear whether Sarah interpreting "yes" as "probably" would simply be a matter of mislabelling or mischaracterising an estimate. For our work, care will be taken to define precisely how the meanings of beliefs are related to the actions of agents holding them (see section 1.22).

If there are restrictions on how a question can be answered and there is uncertainty about which answer is strictly true, estimation may become relative to interests. When communicating beliefs there are often restrictions on how much information can be communicated, or on the language that they are communicated in. Similarly, the human mind has limited memory, conclusions formed in day-to-day life may also need to be abbreviated. A restricted set of possible states of a mind can be seen as analogous to the restricted language of allowed answers from the bank example.

The bank example is described as communicating a complex belief by abbreviating

and distorting it in such a way that the new belief is still useful to the person being communicated to. The constraints force distortion of beliefs, what the best distortion is depends on how the inference is intended to be used.

The process by which a single agent transitions its beliefs over time may also be thought of as communicating to its future self. As an example, imagine that you have once travelled from Melbourne to Sydney in the back seat of a car. Later you will have to drive back alone without GPS, a map or a photographic memory. It will be necessary to abbreviate the memory of the journey so that it will include only some important information, e.g. landmarks and difficult intersections.

When given new information, one wants to adjust one's beliefs to incorporate it. Unfortunately, for real agents transitioning beliefs in real environments, there will be constraints on what beliefs are allowed. This thesis will demonstrate that the concept of transitioning beliefs under constraints guided by the intended application is a useful tool for analysing inductive inference.

1.2 Statistical Example

The previous section gave some informal examples of interest-relative inference. We now give a simple statistical example using a modified Bernoulli trial example. A Bernoulli trial is a random experiment with exactly two possible outcomes, 1 and 0. The probability of outcome 1 is the same every time the experiment is conducted. This can also be imagined as repeatedly tossing a biased coin with 1 being heads and 0 being tails.

For our modified problem, the coin will become more and more biased each time it is tossed. Let $z = (z_1, z_2, \dots)$ be the trial outcomes where $z_i \in \{0, 1\}$ for all i. Let $b \in [0, 1]$ denote the initial coin bias. Define the probability $f(z_i|b)$ of trial i having outcome 1 when the initial bias is b as, $f(z_i|b) = \frac{1}{2}(2b)^i$ when $b \leq 0.5$ and $f(z_i|b) = 1 - \frac{1}{2}(2(1-b))^i$ when $b > 0.5$. If the initial bias is greater than 0.5, it will increase with each consecutive trial (e.g., $f(z_1|0.6) = 0.6$, $f(z_2|0.6) = 0.68$ and $f(z_3|0.6) = 0.744$). If the initial bias is less than 0.5, it will decrease with each consecutive trial (e.g., $f(z_1|0.3) = 0.3$, $f(z_2|0.3) = 0.18$ and $f(z_3|0.3) = 0.108$). This problem will be useful for illustrating our point because the cost of making a bad estimate increases the further into the future one wishes to predict.

Let $x = (z_1, z_2, \ldots, z_N)$ be a sequence of observed trial outcomes. Suppose we wish to make inferences about initial bias b given observation x. A Bayesian approach will be used by assuming a uniform prior $h(b) = 1$ over b.

An inference is now to be made. The premise of this inference is composed of: the assumed prior h, the likelihood function f, and the observation x. From this premise a conclusion is to be derived. Imagine that there are two agents which have two different interests. Call these agents A_1 and A_2.

Agent A_1 will be asked to predict the outcome of the next experiment $y = (y_1) = (z_{N+1})$ by specifying a probability for each possible outcome of y. It will be penalised by a quantity of utility proportional to the logarithm of the likelihood of its answer given the actual outcome of y. The behaviour of A_1 must follow from the conclusion it is given. It must behave as would be optimal if the conclusion given it were true.

Agent A_2 will be asked to predict the combined outcomes of the next two experiments $y' = (y_1, y_2) = (z_{N+1}, z_{N+2})$ by specifying a probability for each possible outcome (there are four) of y'. It will be penalised by a quantity of utility proportional to the logarithm of the likelihood of its answer given the actual outcome of y'. As an example, imagine that the value of y' turned out to be $(0, 1)$ while the agent had assigned probability 0.2 to that possible outcome, the utility reward would then be $\log(0.2)$.

This type of log-likelihood penalty is a commonly used measure of predictive performance. It has been argued (see [Dowe, 2008, footnote 175], [Dowe, 2011, sec. 3] and [Dowe, 2013, sec. 4.1]) that log-loss is preferable as a go-to method of predictive scoring as it is uniquely invariant to how the question about what is to be predicted is phrased. We are not suggesting here that log-loss is the only acceptable loss measure, but simply that it is sufficiently common and motivated for use in our example (the example should still work with other measures).

In the previous section (1.1), we saw that interest relativity can arise when the language of allowed conclusions is restricted. Let us consider two different languages of conclusions for this problem and denote them by L_1 and L_2.

Define L_1 as the language that requires a single estimate of the parameter b. Let L_1 be the set of all statements of the form "$b = v$" where $v \in [0, 1]$. If the conclusion given to A_1 is the statement "$b = 0.4$", its prediction must then be, $P(y_1) = f(z_{N+1}|0.4) \wedge P(\bar{y}_1) =$

$f(\bar{z}_{N+1}|0.4)$. A_1 must act (here predict) as if its given conclusion "$p = 0.4$" is true. As a notational shorthand, $f(y_i)$ is used to mean $f(y_i = \mathit{true})$ while $f(\bar{y}_i)$ is used to mean $f(y_i = \mathit{false})$.

Next, L_2 is defined as the set of all statements of the form "$P(b = v_1) = 0.5 \wedge P(b = v_2) = 0.5$" where $v_1, v_2 \in [0, 1]$. This is the language that makes two separate estimates of the parameter b and assigns equal degrees of belief in each. Such multi-estimate languages are not typically used for simple one-parameter problems, but are common for more complex problems where Monte-Carlo search algorithms (randomised search algorithms) can be used to produce multiple estimates which, when combined, can make better predictions. Note that L_2 generalises L_1: for any member of L_1 there is an equivalent member of L_2. If the conclusion given to A_1 is the statement "$P(b = 0.3) = 0.5 \wedge P(b = 0.4) = 0.5$", its prediction must be,

$$\begin{aligned} \text{“ } & P(y_1) = 0.5f(z_{N+1}|0.3) + 0.5f(z_{N+1}|0.4) \wedge \\ & P(\bar{y}_1) = 0.5f(\bar{z}_{N+1}|0.3) + 0.5f(\bar{z}_{N+1}|0.4) \text{ ” .} \end{aligned}$$

Similarly, if the same conclusion were given to agent A_2, its prediction must be,

$$\begin{aligned} \text{“ } & P(y_1, y_2) = 0.5f(z_{N+1}|0.3)f(z_{N+2}|0.3) + 0.5f(z_{N+1}|0.4)f(z_{N+2}|0.4) \wedge \\ & P(y_1, \bar{y}_2) = 0.5f(z_{N+1}|0.3)f(\bar{z}_{N+2}|0.3) + 0.5f(z_{N+1}|0.4)f(\bar{z}_{N+2}|0.4) \wedge \\ & P(\bar{y}_1, y_2) = 0.5f(\bar{z}_{N+1}|0.3)f(z_{N+2}|0.3) + 0.5f(\bar{z}_{N+1}|0.4)f(z_{N+2}|0.4) \wedge \\ & P(\bar{y}_1, \bar{y}_2) = 0.5f(\bar{z}_{N+1}|0.3)f(\bar{z}_{N+2}|0.3) + 0.5f(\bar{z}_{N+1}|0.4)f(\bar{z}_{N+2}|0.4) \text{ ” .} \end{aligned}$$

For this example: does what the best conclusion is, depend on which agent will receive it? The answer can be seen easily by considering a concrete case. Imagine that the observed data is $x = (1, 1, 1, 0, 0)$. The best estimates are shown by table 1.1. The corresponding expected utilities will then have the values shown by table 1.2.

Expected log-loss was used as the value to optimise. One benefit of minimising expected loss is that, if the inference were hypothetically repeated many times under the same conditions (assuming the true data source imitates the prior), expectation is the measure that will lead to the best total long-term utility. This justification is not perfect as many inferences are one-off occurrences and there are other objectives that could be optimised (e.g. worst case utility). Nonetheless, an objective must be chosen to give a concrete example.

Table 1.1: optimal estimates for each agent-language pair

	L_1	L_2
A_1	"$p = 0.4827$"	"$P(p = 0.4575) = 0.5 \ \wedge \ P(p = 0.5028) = 0.5$"
A_2	"$p = 0.4836$"	"$P(p = 0.3957) = 0.5 \ \wedge \ P(p = 0.5365) = 0.5$"

Table 1.2: expected utilities for the optimal estimates

	L_1	L_2
A_1	-0.67499	-0.67499
A_2	-1.34779	-1.28440

Notice that the best estimate differs according to the agent and the language of allowed conclusions. The more flexible language L_2 is not needed for the agent (A_1) which predicts only a single trial – it improves expected utility by a tiny amount (not noticeable within 5 decimal places). For the agent (A_2) which predicts the next two trials together, the more flexible language L_2 does give a decent increase in expected utility. The numbers from tables 1.1 and 1.2 were found by searching L_1 and L_2 using intervals of 0.0001 for p. This numerical solution was used because minimising expected Kullback-Leibler divergence tends to be mathematically tedious and – in this case – writing a brute-force algorithm was easier and adequate.

As demonstrated in the previous section (1.1), there are two obvious ways to reduce or eliminate interest-relativity. The first is to permit a less constrained language of conclusions. If the language of conclusions is the set of all probability distributions defined over the parameter b, then the conclusion can be set to the posterior distribution. Both A_1 and A_2 could derive optimal behaviours from that conclusion and it is the same for both.

A second way to reduce the interest-relativity is to gather more data. With data $x = (x_1, x_2, \ldots, x_N)$, as N approaches infinity, the best conclusions for both A_1 and A_2 can be expected to approach each other. This happens under both L_1 and L_2 – though, it requires more data for L_2 than L_1.

Given that these ways of reducing interest-relativity exist, it might be suggested that interest-relative induction is a form of reasoning only for less-ideal circumstances. It is nonetheless, a form of reasoning that can be justified. For example, the L_1 conclusion

language constraint could be justified if the agent that is to use the conclusion can only understand conclusions of that form. We will argue that generally, when concerned with inference for producing explanations, such constraints do apply and are significant enough to affect how inference should be performed (see section 3.1).

1.3 Motivation

To recap, for inference to have a *realistic context*, there must be an agent which performs the inference and an agent which uses the conclusion produced. The term 'interest' is used to refer to the practical investment and limitations of these agents. Practical investment is what the agents want, and practical limitations are what they are and are not capable of.

The idea that normative beliefs (stating what is desired) should influence the formation of positive beliefs (stating how things are actually believed to be) is not one that naturally appeals to statisticians or researchers. A belief that is formed to be useful for one's own purposes only, is a self-centred belief. It is preferable that positive beliefs – especially in science – have the potential to serve many purposes. Intuitively, interest relativity does not seem desirable. The problem, as we see it and will demonstrate, is that practical limitations exist for people and programs and that these make interest relativity unavoidable in many cases. To know how to deal with this effect, it is necessary to understand the relation between interest and inductive inference.

As the previous two sections (1.1 and 1.2) have demonstrated, it is easy to find examples of interest influencing how inferences should be made. There is however, a difference between demonstrating that the relation exists through examples and understanding it in a general sense.

In some branches of philosophy this topic has been considered in more general sense. Examples include [Kuhn, 1962] in philosophy of science and Stanley [2005] in epistemology. We believe that the idea is currently not adequately defined or explored in a way that is compatible with, and easily applicable to, fields directly concerned with designing inference algorithms. Examples of such fields include: Bayesian statistics, machine learning, data-mining, artificial intelligence and decision theory. In these fields, formulations typically capture a specific interest but are not concerned with representing the range of possible interests.

There is some work which goes partially towards filling this gap. One example is Bayesian estimators which represent the practical investment as utility via loss functions (see section 1.17). Our work demonstrates why these approaches fall short of adequately representing the relation. The two main reasons are: Firstly, existing work does not attempt to delineate the possible range of valid *practical limitations*. Our work suggest a broad definition (definition 9) and also a candidate formal definition (sections 2.5 and 2.10). Secondly, separation of prior belief and environments is needed to analyse interest-relativity (section 1.20) and this is often overlooked.

A comprehensive analysis of the role of interest in inductive inference is needed using methods applicable to fields directly concerned with designing inference algorithms. The concepts needed for an understanding of this topic need to be identified and developed.

1.4 Scope and Method

This thesis is primarily intended to address a gap in the theory underlying the field of practical inductive inference method design. Currently, this includes methods based on Bayesian probabilities or algorithmic probabilities – methods which appeal to hypothesis simplicity and/or prior belief. The way concepts are formulated in this work reflects that focus. The topic of this thesis overlaps somewhat with philosophy of science, philosophy of mind and epistemology. We try to maintain compatibility and to acknowledge the overlap.

There are many ways to approach the goal of developing a better understanding of interest-relative inductive inference. For example, existing practice in data analysis could be examined to identify the different types of interests that are commonly used. Advocates of the minimum description length principle [Rissanen, 1978, Grünwald, 2007], for example, have 3 different approaches for three different interests: prediction, model selection, and parameter estimation. Shmueli [2010] has, by looking at different approaches to experiment design and data analysis existing in practice, concluded that prediction and explanation are distinct interests requiring different approaches. Going further, they make arguments for why the divide between prediction and explanation as separate interests must exist. Similarly, Dowe [2011, sec. 4.1] (and in [Dowe, 2013, sec. 4.2]) has described the difference between Bayesian prediction and explanation methods.

An "empirical" approach to our goals, using existing practice as its data, might help identify the different types of interests that are currently typically adopted in data analysis. It must be noted, as Shmueli [2010] does, that these different methods have come about often without any explicit acknowledgement of the role of interest. An approach to this topic that focuses on exploring existing practice could be useful; but, without daring to imagine hypothetical alternatives, there is a risk of reinforcing existing tacit assumptions.

It is desirable that theories be tested by looking at the predictions that they make and by then setting up experiments that may contradict or support them. For theories concerning the nature of inductive inference, such experiments may be hypothetically possible but are simply not practical. One could imagine giving one approach to data analysis to one set of researchers and another to a second set and then waiting to see which group produces the best research, but this is unrealistic.

In machine learning and statistics, a common method for defending an approach is to show that it performs well when applied to many real world data sources. Interest determines the measure of performance that should be used. In machine learning it is common to use predictive performance as "the" measure of performance, but there are many ways in which performance might be measured and many measures of predictive performance. The reader must understand that while this thesis is intended to apply to machine learning and statistics, our goal is not to promote or introduce a specific inference algorithm or family of inference algorithms. We want to look at how the measure of performance itself is justified. For this reason the methodology that is appropriate for our topic is closest to philosophy of science.

Our approach begins by proposing a clear definition for the concept of interest-relative induction. A small set of simple premises/axioms/assumptions is then fixed so deductions about the properties of interest-relative induction can be made. These assumptions are listed in section 1.23 under the heading "primary assumptions". It is then argued that any non-interest-relative account of inductive inference will fail to account for the need for, and justification of, the types of inference methods that are considered typically inductive. The remainder of the thesis explores the implications of these assumptions. The main tool for eliciting these implications is a formal model developed in chapter 2. Our model attempts to precisely define the set of all possible

interests that may apply to a given inference task. It is designed to be as general as possible. It is designed to describe without ambiguity the effect of practical investment and limitations on induction.

1.5 Thesis Overview

In this chapter (chapter 1), our assumptions are carefully argued for. The implications of these assumptions are then explored in the remaining chapters.

In chapter 2, a formal model of interest and how it relates to inductive inference is developed. Some simple theorems for the model are produced. These are then used to elicit some general principles for interest-relative induction.

In chapter 3, some topics important to the fields of machine learning, statistics, artificial intelligence and algorithmic information theory are discussed from the perspective of interest-relative induction.

In chapter 4, an interest-guided approach to inductive inference is proposed. Decision tree problems are used to demonstrate how being armed with some simple principles can help make the process of designing inference algorithms easier and make the end product more effective.

Chapter 1 is designed to be accessible to readers with a basic knowledge of probability, logic and set theory. Knowledge of Bayesian probabilities, algorithmic probability and decision theory would also help. This chapter should be read first by all readers and all sections in order. The literature review at the end of this chapter could be skipped without risking loss of understandability of subsequent chapters.

Chapter 2 is much more technical and requires a good feel for abstraction. Readers wanting to avoid it are recommended to skip to the chapter summary (section 2.19).

Chapter 3 covers several topics and it is not necessary that all its sections be read. It is recommended that at least the preceding chapter summary (section 2.19) has been read first.

Chapter 4 assumes some familiarity with Bayesian inference methods and specifically a basic knowledge of decision trees.

1.6 Propositions, Beliefs and Inference

This section briefly covers the concepts and terminology of propositions, beliefs and inference. A *proposition* is some statement. A proposition may have a truth value which is either true or false. Under some systems of logic propositions may exist which can be neither be true or false.

There are different approaches to defining what beliefs are, we will follow an approach that derives belief from behaviour. Such approaches are broadly known as "behaviourism". According to that interpretation, a belief is a predisposition to act/behave in a certain manner. Different behaviours imply different beliefs. For our purposes, beliefs are held by agents about propositions. An *agent* might, for example, be: a person, a computer program, an animal or some collective of these.

Behaviourism is not always the ideal approach to defining belief. It is less popular now in psychology and neuroscience. Our work is concerned firstly with formal notions for inference in: science, artificial intelligence, machine learning, and statistics. For our purposes, it is more important how an agent's belief relates to its actions and communications than what its internal state is.

There are many systems for formally describing beliefs. Under *all-out belief* systems, a given proposition may be believed to be true, believed to be false or no belief about it might be held. Under *partial belief* systems, it is possible to hold both a proposition and its negation as possible – belief may come in degrees. Holton [2008] gives more precise definitions of these concepts (all-out and partial beliefs) and how they relate to an agents willingness to act.

If the degrees of belief must come from the real interval $[0, 1]$, the belief is called a *credence*. Partial beliefs as credences can be manipulated as probabilities and this is how *Bayesian beliefs* (Bayesian probabilities) work. In [Ramsey, 1931], credences are used to formulate subjective beliefs; these credences are derived from the hypothetical betting behaviour of agents.

Inference is concerned with producing a *conclusion* from a *premise*. Both premise and conclusion are beliefs and may be attributed to some – possibly hypothetical – agent. Formal systems defining how beliefs can be represented are known as *logic systems*. Formal logic systems include rules of inference which define how conclusions may be

safely derived from premises. If a conclusion is produced by strictly following these rules of inference then the inference can be called *deductive*.

As an example, a premise might be composed of the two following propositions: "*all crows are the same colour*" and "*I have observed 10 crows and all were black*". A deductive inference might then produce the conclusion, "*all crows are black*".

Some might make a distinction here between the two propositions composing the premise by noting that the first is a universally qualified statement while the second is not. These are sometimes referred to as major and minor premises respectively. The major premise is a general statement while the minor premise is a specific statement.

In section 1.2 the reader might have noticed that we were using the terms 'premise' and 'conclusion' where statisticians might write data and estimate. The difference between a conclusion and an estimate is that an estimate "estimates" some parameter. For the example from that section, the proposition "$p = 0.3$" is a estimate of parameter p. The proposition "$\Pr(p = 0.3) = 0.5$" is not an estimate of p and there might be no variable, parameter or event that it could be said to estimate. The statement "$\Pr(p = 0.3) = 0.5$" can be a conclusion if it is a valid belief under the chosen system of logic. Similarly, premises and data are not the same thing. The prior distribution $h(p)$ from the example in section 1.2 is not data, nor is the proposition "*all crows are the same colour*". Both of these beliefs may be considered premises but not data.

We use the term 'premise' to group together assumptions and data – general and specific. It should be noted that under some systems of logic, like those concerned with probabilities, it might not always be clear how to divide all statements into these two categories (general/specific or major/minor).

Reasoning that destroys previously held beliefs is known as *destructive*. Most commonly used systems of logic – such as predicate logic and propositional logic – do not have rules of inference that may remove previously held beliefs. For this reason it is often assumed that what is believed after an inference is the conclusion and the premise. For our goal of looking at inductive inference, and specifically interest-relative induction, it is necessary that this assumption be reversed. It is assumed that the conclusion will be passed on to some agent which might not know the premise. If the premise, or part of it, is to be passed along with the conclusion, the conclusion will have to explicitly contain it. The properties of being destructive or non-destructive are also sometimes referred to

as non-monotonic and monotonic respectively.

Just as an inference can destroy information, it may also create information. Those all-out systems of logic that most readers are likely to be familiar with – such as predicate or propositional logic – tend not to have rules of inference that may produce new information. Their rules of inference only produce new statements that are already entailed by the given premise. Reasoning that does create information is known as *amplicative* (or sometimes *ampliative*). Amplicativity is typically a property associated with inductive inference. Here is an example: The premise is, "*I have observed 10 crows and all were black*". The conclusion is, "*all crows are black*". Note that this time the statement that all crows are the same colour was not part of the premise. The conclusion does not follow deductively from the premise. It may be that the 11th crow is gray. This inference creates (assumes/induces) information and is amplicative.

Where all-out logic systems have inference rules that produce new propositions from a set of already believed propositions, when reasoning with credences it is required that the credences be manipulated as probabilities – conform to the probability calculus. It was shown by Kemeny [1955] that no Dutch-book (exploiting incoherent betting behaviour) can be made against an agent if its beliefs conform to the probability calculus. With this requirement, beliefs as credences can be coherent in its relation to behaviour.

Bayesian probability, as opposed to other interpretations of probability, considers probabilities to be degrees of belief held by some agent about propositions. This is different from the frequentist interpretation where probabilities are properties of events, or algorithmic probabilities which are derived from description structure. Since interest is something that must be attributed to agents, it makes sense to approach our topic from a perspective that is concerned with beliefs which can be held or attributed to agents.

1.7 Observable and Unobservable

The entities that beliefs are concerned with are sometimes divided into two classes: *observable* and *unobservable*. An observable is something that may be directly observed through some experiment.

It is necessary to note that whether one considers something like a proposition to be observable, depends on an entire interpretation of the world. For example, to say that a black crow was observed requires that the concepts "black" and "crow" and how they may be identified must already exist. This is known as the theory-laden nature of observation and has been used to argue against absolute empirical interpretations of science. For our work, this is not a question that will be focused on. We will assume that all propositions may be labelled as observable and unobservable – it would be difficult to discuss inductive inference without making such a distinction. We recognise that the line between observable and unobservable is not an absolute one; but the details will be considered out of scope for this thesis.

A further distinction must be made between two separate motivations for reasoning about an unobservable. An agent might reason about an unobservable entity because it is actually believed to exist, or, because it is considered a useful tool for linking observables (making predictions).

Scientific realism is an interpretation of the scientific method which holds that scientific theories are intended to uncover the actual unobservable entities of the world. By contrast, instrumentalism is an interpretation of the scientific method which holds that scientific theories are concerned with unobservables only as useful tools. Questions about whether they actually exist or not are considered meaningless.

For our purposes, a proposition will be considered unobservable if it is only a construct in the implementation (mechanism) of an agent's mind used as an intermediate for reasoning about things that can be directly measured. Hypothesis selection and parameter estimation problems are seen as inference to unobservable propositions while prediction concerns observable propositions. In this thesis, when considering inferences producing conclusions which make statements about unobservable entities, it is assumed that the value of those unobservable-concerned beliefs lie only in how they aid behaviour decisions for agents in environments which can be defined exclusively in terms of observables. Unobservables cannot be directly of interest but may be indirectly (this is discussed in more detail in section 1.17).

1.8 Probability Spaces and Notation

To be able to talk about probabilities more clearly it is necessary to describe here what probability spaces are and what is needed for a probability space to be well defined. For a probability space to be defined, a set of elementary events Ω is first defined. Elementary events are fully instantiated possible states of the world. If it is known which elementary event the state of the world is in, there is nothing more to learn. Elementary events can be grouped together to form compound events. We will refer to compound events simply as events. Let $\tau \in \Omega$ denote the true state of the world. If $e \subseteq \Omega$, believing compound event e is equivalent to believing $\tau \in e$.

If Ω is countable then the set of all compound events E is simply the power-set of Ω (the set of all subsets). If Ω is uncountable then a σ-algebra – a set of well defined subsets – must be specified. This is because if a positive probability is defined for uncountably many elementary events then their probabilities would sum to infinity. A σ-algebra defines a set E of well defined subsets of Ω to avoid this.

We will not go into the details of how a σ-algebra is defined since that would be a long technical detour. The origin of measure-theoretic probability can be found in [Kolmogorov, 1950] but the reader can find more accessible introductions through encyclopedia entries for "probability space". When discussing the uncountable case we will avoid mentioning the form of the σ-algebra when possible for the sake of brevity. It should be assumed to be there and defined.

A probability function is a function mapping all events from the set E to the interval $[0, 1]$ of the real numbers. The probability of event $\Omega \in E$ must be one. The probability of the union of two disjoint events must be equal to the sum of their individual probabilities. Note that some elements of, or subsets of, Ω might not be events (not members of E) – in which case they have no probability value defined.

A probability space is composed of an elementary event set Ω, a σ-algebra over Ω with event set E, and a probability function P defined over all members of E. $P(e)$ denotes the probability of event $e \in E$.

A random variable of a probability space is a function $f : \Omega \to R$ where R is some set. The preimage of a random variable defines a partition Z of the set Ω. For this work, that partition will be referred to as the random variable; our concern is more with how

it partitions Ω rather than the actual function.

As a notational convention, upper-case letters will be used to denote partitions of Ω as random variables. Lower-case letters will be used to denote specific events. E.g., Z is a random variable and a set of events; $z \in Z$ is a specific element of that set (an event $z \in E$). $P(z)$ represents a single real valued number (the probability of event z) while $P(Z)$ represents a distribution over the possible values of random variable Z.

The notation $P(Y|\theta)$ will be used to denote the conditional probability distribution over variable Y given event θ. If, instead, θ is not an event or variable of E but a parameter of the probability function, $P(Y;\theta)$ will be written.

A dot will be used to denote a probability function not restricted to a specific event or variable, so $P(\cdot|e)$ represents the distribution of all events conditional on $e \in E$.

1.9 Our Approach to Belief

The approach that we take to defining beliefs is somewhat complicated. Primarily, the method of deriving credences from hypothetical behaviour as used by Ramsey [1931] in defining subjective beliefs is followed. We generalise this method first by allowing behaviour to be undefined for some hypothetical decisions. This leads to the possibility of vague beliefs and the use of what is known as *imprecise probabilities*. We also complicate the relation between belief and action by imagining that some agents do not behave optimally for their beliefs. For some readers this deviation may seem tedious. The motivations for it are given in this section; they center around the need to be able to discuss limited agents (limited in reasoning abilities). Section 1.22 describes in more detail our method while sections 2.3 and 2.5 give the exact definition.

Human beings cannot really be said to explicitly reason with credences; nor is our reasoning clearly all-out as with propositional logic. Humans are more comfortable with qualitative probabilities and tend to perform inconsistently when trying to use explicit probabilities [Budescu and Wallsten, 1995, Kahneman and Tversky, 2000]. We tend to overestimate highly improbable events and underestimate highly probable events.

This creates a serious obstacle for analysing the concept of interest-relative induction as it is exactly the imperfect nature of agent minds that motivates our topic. If one

tells an agent that proposition p has probability $\Pr(p) = 10^{-9}$ and it behaves as if $\Pr(p) = 10^{-5}$, an attempt to directly derive belief meaning from behaviour would say that it believes $\Pr(p) = 10^{-5}$ because it acts as if that was true. It might, however, be known that the agent simply acts wrong because its limited mind cannot deal well with very small probabilities.

A distinction must be made between explicitly held beliefs and implicitly held beliefs. I might explicitly believe propositions p and $p \Rightarrow q$ while only implicitly believing q. For belief in q to become explicit, an inference producing q must be explicitly performed.

Holton [2008] argues that our beliefs are not always credences but are also not always all-out. It is argued by Holton [2013] that we tend to explicitly hold a small number of all-out possibilities. This set does not contain all possibilities, only those which we intend to work with. Such a small set of explicitly held all-out beliefs (live possibilities) is called a *running belief*.

The motivation for running beliefs is easy to understand when considering how belief must be turned into behaviour. If a joint probability is defined over a set of N separate propositions with all combinations having non-zero probability, then 2^N possible world states must be considered. Calculating what actions are best becomes impractical. This leads Holton [2008] to suggest that credences are the partial beliefs that agents would hold if they were unconstrained by practical limitations.

Our approach to formulating belief is concerned with limited agents. From the above examples, two general statements about limited agents can be elicited: First, the set of beliefs that a limited agent may hold can be restricted. Second, for the environment that a given limited agent is intended for, that agent might not always be able to determine what the best behaviour to adopt is even when the belief it holds entails one because it cannot explicitly consider all implications of its beliefs.

A second problem is that limited agents may hold vague beliefs; beliefs that recommend actions for some hypothetical decisions but say nothing about others. Imagine now that E is an event set representing the observable world – i.e., each $e \in E$ is an observable proposition. A Bayesian belief for world E takes the form of a single probability function P over E. Here, $P(e)$ is defined for every event $e \in E$.

In section 2.3 we introduce a class of beliefs which we call *incomplete Bayesian beliefs*

or simply *incomplete beliefs*. This class is based on Bayesian beliefs but differs in that P does not always define an exact value for $P(e)$ for each $e \in E$. Incomplete beliefs differ from Bayesian beliefs in that they allow more vague beliefs. We will use the terms *complete Bayesian beliefs*, *complete beliefs*, or *Bayesian beliefs* when referring to the traditional non-vague variety. Our incomplete beliefs use what is known as imprecise probabilities where complete beliefs use precise probabilities. An imprecise probability is a set of probability functions. As an example, let p be some proposition; the belief $\Pr(p) = 0.7$ is a precise probability, the belief $\Pr(p) > 0.5$ is an imprecise probability – it can be represented as the set of all probability functions conforming to that inequality. We use the term "incomplete beliefs" to refer to our method for relating imprecise probabilities to hypothetical behaviour using a method similar to how subjective beliefs are derived from hypothetical betting behaviour by Ramsey [1931]. For other approaches to relating imprecise probabilities to dispositions to behaviour see [Walley, 2000].

Section 1.22 describes in more detail our method of relating belief to action for limited agents. The idea that rationality can be defined for limited agents according to behaviour constraints is not new (see [Baron, 2005]). Our work goes further by introducing a method for defining shared meaning among limited agents having possibly different limitations. Section 2.3 gives the exact definition for incomplete beliefs. Section 2.5 defines an exact relation between imprecise probabilities and the behaviour of limited agents.

The remainder of this chapter describes many concepts using examples which employ complete Bayesian beliefs. The reader should keep in mind that this is done to keep the discussion simple and that many of these concepts are meant to apply to languages of belief in general.

1.10 Inductive Inference

Some common interpretations of what inductive inference is are described in this section; there is currently no consensus on what the best exact definition is.

Deductive inference is usually simpler to define, it is reasoning that only produces conclusions which are entailed by the given premise under the chosen system of logic. For predicate logic, new propositions can be derived from known propositions using

repeated application of rules of inference. For Bayesian beliefs, degrees of belief in different propositions can be related using the probability algebra deductively.

Inductive inference is much harder to define. While the term 'inductive inference' is hard to define in the context of all-out beliefs, the definition of inductive inference in the context of partial beliefs and credences seems even more unclear. Unfortunately, use of this term in the context of credences is so loose in the literature that to try to conform to all interpretations would make it lose all meaning.

We will divide interpretations of what inductive reasoning is into three broad categories which are summarised as definitions 1, 2 and 3. These three might not capture the whole range of existing interpretations, but they do give some overview of how existing interpretations differ in essential ways.

Definition 1. *Inductive reasoning is any reasoning that is not deductive.*

Definition 2. *Inductive reasoning produces conclusions about unknowns given knowns.*

Definition 3. *Inductive reasoning produces generalisations from specific instances.*

Under definition 1, it is not necessary that the conclusion say something general that was not in the premise, nor does it even need to make any exact prediction. Simply changing what was in the premise is sufficient; for example, by adjusting the degree of belief about some specific entity, or by forgetting something that was already believed. The inference does not necessarily need to make additional (not contained in the premise) statements about unobservables.

Under definition 2, again the inference does not need to make additional statements about unobservables. An unknown could simply be an unobserved observable. Here, any prediction is inductive.

Generalising is sometimes considered a key characteristic of inductive inference. A traditional example of the concept of generalisation, for all-out reasoning, is to imagine that 10 crows have been observed and all were black, it is then concluded that all crows are black. Here the 10 crow observations are specific instances and the conclusion that all crows are black is a generalisation. Here the general and specific are analogous to the observable and unobservable from section 1.7.

Note that under definition 2, it is sufficient to make a prediction about some specific observable. For definition 3, this alone would not be induction; a generalisation must be made, some belief about an unobsevable must be altered.

Applying definitions 1, 2 and 3 (of inductive inference) to the Bayesian context presents a problem. Bayesian inference methods are typically concerned with transitions in belief triggered by the acquisition of observational data. One starts with some prior belief, gains new data and then moves to a new belief. In the next paragraph, an example will be used to demonstrate the problem. Note that some authors present Bayesian updating – applying Bayes rule to select a hypothesis – as "the" Bayesian inductive inference method. This can lead to confusion. A Bayesian inference method is simply one that employs the Bayesian interpretation of probabilities.

Let us imagine that the probability $P_0(e) \in [0, 1]$ represents our prior belief about event of interest e. Here e is an observable entity that we wish to make a prediction about. Now event x is observed. Let $P_1(e) = P_0(e|x)$ be the posterior probability of e given evidence x. Let $P_2(e)$ be our inferred belief about e; here P_2 is our conclusion.

In the case where the inferred conclusion is not equal to the posterior $P_2(e) \neq P_1(e) = P_0(e|x)$, the inference must be distorting belief and would seem to be inductive (under definition 1). Note, a precise definition for "distorting belief" will be given in section 1.12. This property is common among inference methods which estimate intermediate parameters. Imagine for example that we use the prior P_0 and the evidence x to estimate some intermediate parameter θ. Then we use $P_2(\cdot) = P_0(\cdot|\theta) \neq P_1(\cdot) = P_0(\cdot|x)$ as our inferred belief, in which case $P_2(e)$ only attempts to approximate $P_1(e)$. Such inferences of intermediate parameters are typically described as inductive and are often amplicative as well. The amplicativity here would arise when evidence x does not give enough information to narrow down uncertainty about θ to exactly one possibility while the inductive method selects an exact value.

For pure Bayesian prediction, the posterior probability of the event of interest is used as the conclusion with no intermediate parameter estimations. In that case, $P_2(\cdot) = P_1(\cdot) = P_0(\cdot|x)$. It is now unclear whether this is a case of inductive reasoning or deductive reasoning. Inference is concerned with moving from a premise to a conclusion. If both prior P_0 and evidence x are considered part of the premise, then P_1 (the posterior distribution) is the premise; hence, the conclusion $P_2 = P_1$ cannot be inductive.

Our question is unfortunately not so easily answered. An alternative interpretation may be taken. One might argue that by assuming some prior distribution P_0, the inference becomes inductive because P_0 was not reached by deduction but was used "inductively" so that an inference could be made. Under this interpretation only the evidence x is part of the premise (P_0 is not part of the premise but part of the inference method) and using the posterior as the conclusion is then a case of inductive reasoning.

If the prior belief is considered part of the premise then by definitions 1, 2 and 3, the use of prior belief by itself does not make an inference inductive. Alternatively, if the prior belief is not considered part of the premise – but rather a part of the inference method – then the use of prior beliefs can make inferences inductive.

Whether prior belief – such as is employed by Bayesian methods or Solomonoff induction [Solomonoff, 1964, 1996] – should be considered part of the premise or part of the inductive method, does not seem to be a question that has a clear answer. For our work, it will be assumed that for a given inference, both the prior beliefs and observations which are received or assumed by the inference method, are part of its premise. When we use the term 'premise', it will be specifically to indicate that we are talking about more than just observed data.

1.11 Prior Beliefs

For inductive inference, degrees of belief are often more useful than all-out beliefs. It is desirable for evidence to increase or decrease credence in a given hypothesis, instead of producing certain conclusions prematurely, or over-cautiously producing empty conclusions.

The two dominant formal paradigms for applied inductive inference are: those that employ the Bayesian interpretation of probabilities, and those that employ the algorithmic interpretation. The Bayesian interpretation derives credences from hypothetical agent betting behavior. The algorithmic interpretation derives credences from the structural simplicity of data strings. Solomonoff [1996] does state that algorithmic probability can be thought of as "Bayesian" method – in the sense that the choice of universal prefix Turing machine should be thought of as specification of *a priori* belief, not in the sense of deriving probabilities from betting behaviors. Both approaches require that some *a*

priori weighting over the possible states of unknown entities be used. Such weightings are often said to be defined over distinct hypotheses.

It is now necessary to ask, what are hypotheses? A clear definition will not be proposed but the next three paragraphs describe some considerations.

From an empirical perspective, a hypothesis must connect predictions about the outcomes of possible (but not necessarily realistic) experiments. Some definitions of hypotheses describe them as candidate explanations of a phenomenon. The claim that all hypotheses are also explanations might be contested. Grouping together predictions in a particular way does not necessarily say anything about the underlying mechanisms that produce a phenomena, nor the causal structure relating variables. For Solomonoff prediction, individual programs might be considered hypotheses, but it is unclear if such programs are explanations in the traditional sense. This question of what makes a hypothesis an explanation will be revisited in section 3.1.

Many distinct causal models can be made to fit and predict particular phenomena in exactly the same way [Pearl, 2009]. Should two hypotheses with exactly the same predictions be considered the same? Imagine that they refer to different unobservable entities which happen to lead to the same predictive model over all observable entities. For Solomonoff prediction, there will be many separate programs that lead to exactly the same predictions.

It is also unclear if hypotheses are necessarily unobservable. The truth of the hypothesis, "*all crows are black*", is unobservable if there are infinitely many crows: infinite observations cannot be made. If there is a finite number of crows, the truth of this "hypothesis" can be observed directly. For Solomonoff prediction, the underlying program can never be observed directly.

We haven't provided any clear answer as to what should and should not be considered hypotheses. The question will be reconsidered in section 3.1. For both Bayesian and algorithmic probability methods, defining hypothesis is part of the inductive method.

Both Bayesian and algorithmic probability methods usually define some set of alternative hypotheses that may connect observables (possible experiments). Both paradigms require that some weighting over the hypothesis set be employed. Because multiple hypotheses can fit a given data set (set of known observation outcomes), weightings over

hypotheses are needed so that predictions from multiple hypotheses can be combined – and to avoid the problem of over-fitting that occurs when only the most likely hypothesis is selected.

For Bayesians, these weightings take the form of a probability function defined over a set of hypothesis. Under the algorithmic probability approach, these weightings are treated as probabilities but the weighting function is a semi-measure. A problem of prior belief occurs here: First, for a set of considered hypotheses to be chosen, something must be known about the phenomena being studied. This requires that prior belief be assumed if an infinite recursion of justifications is to be avoided. Secondly, it must be decided how hypotheses from this set are to be weighted. These considerations together are known as the "problem of induction" [Popper, 2014].

It may seem that simply choosing the largest set of hypotheses possible and using the most evenly distributed prior weighting over them will solve this problem of priors. Unfortunately, this "prior-less" approach leads to meaningless predictions, a problem known as the no-free-lunch theorem [Wolpert and Macready, 1997]. There are, broadly speaking, three popular approaches for solving this problem of prior knowledge – without trying to completely avoid prior knowledge as Popper's approach ([Popper, 2014]) does. These are known as subjective Bayesianism, objective Bayesianism and algorithmic probabilities. These three approaches will be briefly described but the reader should be aware that our descriptions are characterisations used to make simple a range of views.

Subjective Bayesianism embraces the subjectivity of prior belief formulation. A set of hypotheses and a prior probability function weighting them must be chosen and to defend such prior beliefs subjective intuition must be appealed to. Ideally, this means that models are constructed by appealing to expert intuition. Some experts in the given field are asked to describe the different ways that they think the phenomena might work. Weightings over these possible hypotheses are elicited in the same way. Practice often deviates from the ideal here and some less cautious users of Bayesian methods will simply apply their favourite algorithms to all applications with little to no attempt at defending the chosen prior beliefs by appealing to domain knowledge or expert intuition. Note, we aren't claiming that such generic solutions are necessarily bad. They can work if the inference method is flexible and enough data is available – this will be discussed in more detail in section 1.19.

Objective Bayesianism uses what are called colourless priors. A set of hypotheses must still be selected based on domain knowledge – as with subjective Bayesianism – but the weighting over these hypotheses is chosen as to minimise the information assumed about the observable data (see for example [Berger et al., 2009]). This method attempts to reduce subjectivity by making the prior weighting as uncertain as is possible for the given hypothesis set.

The third approach uses *Algorithmic Probabilities* [Kolmogorov, 1965, Chaitin, 1966, Solomonoff, 1964, 1996]. This approach assumes that the observable aspects of the phenomenon can be produced by some Turing machine. The set of hypotheses is built from a universal prefix Turing machine. Each prefix produces a Turing machine and so the set of hypotheses is an enumeration of all Turing machines. The weighting over these hypotheses (prefixes/programs) is chosen somewhat arbitrarily – derived directly from a starting universal prefix Turing machine. This is known as an *algorithmic prior* or *Solomonoff prior*. Note, there are many essentially equivalent ways of formulating one. The choice of weighting is justified – in part – by the fact that a given universal prefix Turing machine can, with the right prefix, emulate any other. This in turn means that the initial choice of universal prefix Turing machine becomes irrelevant once enough data has been collected. Solomonoff prediction has been described as the objective Bayesian method for the set of all computable hypotheses [Berger et al., 2009]. Solomonoff [1996] describes the main sources of subjectivity for prediction using algorithmic probabilities: the choice of universal prefix Turing machine, the choice of observations to include, and the scheme for coding observed data. He asserts there that making these decisions is a form of prior belief specification. What separates the algorithmic probability approach from the subjective Bayesian approach is that the it minimises the role of prior belief and maximises model flexibility.

These three approaches together largely cover the theory that underlies current formal inductive inference methods. Note that aside from these three, under the label machine learning, many methods exist that are inductive but which are not much concerned with theoretical/philosophical foundations. It is preferable that our work does not commit overly to any one of the three main approaches.

The term 'prior belief' will be used throughout this thesis to mean the combination of the assumed hypothesis set and the assumed weighting over it. Note that some authors

use the word 'prior' to refer to only the weighting.

It is not correct to think of a subjective prior as simply something that is made up for practicality. The idea that a subjective prior might actually contain information about the phenomenon being investigated is not unfounded. The person that formulates a subjective prior has information available to them: information obtained from personal experience with similar phenomena, information passed on from other people and information passed on genetically. In [MacKay, 2003], it is argued that the processes of evolution, like learning, can be seen as accumulating information. It might be rash to simply discard the millions of years worth of information that evolution has distilled into the structure of the human mind. Consider, for example, that the human mind has evolved to quickly learn about three-dimensional space, objects, velocity and momentum. This is a form of subjective bias, but it is also useful knowledge about the real world, obtained through natural selection.

Subjective information is, nonetheless, imperfect information. Human beings are subject to all sorts of psychological bias. Some of these – like confirmation bias – do not seem to have obvious justifications and seem more like the result of computational short-cuts. Algorithmic probabilities attempt to minimise the subjectivity of a priori assumptions. Objective Bayesianism does the same to some extent. Increased objectivity comes at a price. For many inference problems, too little data is available for Solomonoff prediction to be practical and even the objective Bayesian priors of Berger et al. [2009] will learn too slowly. Objectivity may be sacrificed by selecting more "loaded" priors – i.e., priors that put much weight on a small subset of possibilities. This allows prediction to be done based on less data but makes the choice of prior harder to justify.

Our model of interest-relative induction introduced in chapter 2 is defined in such a way that any of the approaches to creating priors described above can be applied to them. It should be understood however, that we consider interest-relative induction to be a fundamentally non-ideal type of reasoning. It is not long-term ideal reasoning but a short-term tool for creating beliefs with limited intended application (see section 1.14). We are not concerned with the hypothetical situation of observed data going to infinity. For our purposes, it must be assumed that the priors used are considered to contain enough information about the phenomenon of interest to be useful given the amount of data actually observed.

1.12 Defining Interest-Relative Induction

Enough of the key concepts and terminology of inductive inference has now been covered for us to propose a definition for the term "interest-relative induction". First, we introduce another term, *belief distortion*.

Definition 4. *An inference is* **belief distorting** *iff, for some observable proposition, the conclusion and premise do not make identical predictions.*

For beliefs expressed as Bayesian probabilities, the only conclusions that are not belief distorting are those for which the meaning is the same as the posterior probability $P(\cdot|x)$, where x is the observed data, the dot represents any observable event, and P captures the prior belief. Using the posterior probability for making predictions is known as Bayesian prediction. Solomonoff prediction can be thought of as a form of Bayesian prediction and so is not belief distorting under our definition. To avoid applying the term "Bayesian prediction" to Solomonoff prediction – which is not a Bayesian approach – we will use the term *pure prediction* to refer to those methods that use the posterior probability, or some approximation of, or analogy to it, as the conclusion.

Note that while we have used Bayesian probabilities as an example of a belief language (for premises and conclusions) which definition 4 can be applied to, the definition is meant to apply to any sufficiently well defined belief language.

Many popular inference methods are belief distorting. These include most methods which estimate intermediate parameters or select hypotheses. Note that the way belief distortion is defined here is equivalent to how some define inductive inference. We introduce this term to allow more clarity.

We now propose a definition for the concept that is the focus of this thesis.

Definition 5. *An inductive inference method is* **interest-relative** *iff there is at least one premise for which it will produce a belief distorting conclusion, where the belief distortion is both motivated and justified by a valid interest.*

The word 'iff' is used to mean 'if and only if'. This definition refers to three other concepts which have not been covered yet: *motivation*, *justification* and *valid interests*. Much of the remainder of this chapter will be used to develop those concepts so that

our definition may be complete. For now, the reader can simply think of an interest as the collection of all relevant (to the inference) information concerning the practical investment and limitations of the agents which give the inference method a realistic context. The phrase "*motivates belief distortion*" is used here to mean that the interest tells us why a non-distorting inference method is strictly ruled out. The phrase "*justifies belief distortion*" is used here to mean that the interest tells us why some belief distorting inference method can be acceptable (useful or meaningful).

1.13 Why Define Hypotheses?

Defining and selecting hypotheses is the concern of many inference methods but the motivation for doing this must be considered. This section demonstrates why, for pure prediction, multiple hypotheses need not be defined.

Below is an example of how hypothesis sets are typically defined in Bayesian inference. Note that this example is not presented here to define what hypotheses are in general. Let Ω be our set of elementary events and E the set of events. Assume that E contains only observable events. Let Θ be a set of of hypothesis. Let f be the probability function defined over E given some hypothesis; so the probability of event $e \in E$ given hypothesis $\theta \in \Theta$ is $f(e;\theta)$.

Consider the case where a prediction about unobserved event $y \in E$ is to be made given some observed event $x \in E$. When using pure prediction, a prediction about y can made by summing over the set of all hypotheses Θ using some prior weighting over that set. Let h denote that weighting. The probability of y given x is then,

$$P(y|x) = \sum_{\theta \in \Theta} f(y;\theta) P(\theta|x) \ , \tag{1.1}$$

where,

$$P(\theta|x) = \frac{f(x;\theta)h(\theta)}{\sum_{\theta' \in \Theta} f(x;\theta')h(\theta')} \ . \tag{1.2}$$

An alternative way of looking at this is that a joint distribution $P(E)$ over all events has been defined and it automatically entails the prediction $P(y|x)$. Here, for any

observable $e \in E$,

$$P(e) = \sum_{\theta \in \Theta} f(e|\theta)h(\theta) \; . \tag{1.3}$$

Because $x \in E$, $y \in E$ and $(x \wedge y) \in E$, it follows that $P(y|x)$ is defined. From this perspective the set of hypotheses Θ does not even need to be defined. An alternative hypothesis set Γ with only one member $\Gamma = \{\lambda\}$ can be constructed such that $\forall e \in E$, $f(e; \lambda) = P(e)$.

It can be seen here that if a prior distribution h is defined over a set of considered hypothesis Θ, a single equivalent hypothesis λ exists which defines the same distribution over all observables. For our work, it will be useful to distinguish between such one-hypothesis and multi-hypotheses models. We will call a model a *parameterless model* if it has only one hypothesis. We will call a model a *parametrised model* if it has multiple hypothesis. A parametrised model explicitly defines distinct unobservable entities (the hypotheses). A parameterless model defines an unobservable relation between observable entities, but this relation is not explicitly broken down further by defining distinct hypotheses, sub-models or parameters. For a given parametrised model, there exists an unique effectively equivalent parameterless model – assuming the prior h is defined over all parameters. For a given parameterless model there may exist many effectively equivalent parametrised models.

A single joint distribution over all possible instantiations of the observable events combined is, for pure prediction, all that is needed. This point is noted by Legg [2008] when describing the AIXI agent [Hutter, 2005, Legg, 2008]. From this point of view, Solomonoff prediction simply defines a distribution over the set of all finite strings. The AIXI agent is a hypothetical agent which is *ideal* in the sense that it can learn from any data source and has no computational limitations. An ideal agent acting in some environment needs only predictions. Given predicted probabilities for all unobserved ubservable entities it can always work out how best to act as it has no computational limitations. There is no need for hypotheses from some parametrised model to be selected, except as a computational tool in defining a distribution over observables. The equivalent parameterless model is sufficient and necessary for pure prediction; a parametrised model is not necessary.

Amongst minimum description length (MDL) methods, such pure predictive methods are used in the form of prefix compression codes which are described as having no parameters [Grünwald, 2007].

So why define multiple hypotheses at all? Why make exact inferences about unobservable entities? An ideal agent – one with no computational limitations – can combine predictions and decision theory to behave optimally in its environment. The work of Hutter [2005] and Legg [2008] shows that an ideal agent can eventually learn to act optimally in certain classes of environments even when given no prior knowledge about those environments. A similar approach has been used to define general intelligence [Hernandez-Orallo and Minaya-Collado, 1998, Hernández-Orallo and Dowe, 2010] as a more objective alternative to intelligences tests such as IQ tests. Neither prediction nor intelligent behaviour strictly require that parametrised models be defined. Why then, would an inductive inference method need to make statements about unobservable entities? Why would it need to select hypotheses?

From the perspective of scientific realism (an interpretation of the scientific method, see section 1.7) the answer might be that it is believed that the unobservable entities from a chosen parametrised model actually exist and it is the duty of science to say something about them. One might suggested that statements about unobservables can be used as explanations, they are needed to describe the mechanisms behind a phenomenon or to identify causal relations. This answer – in our opinion – is insufficient: why explain? why identify causes? why not just predict?

A better interpretation is that statements about unobservables may be useful for practical purposes but questions about their actual truth are meaningless. From this perspective, one might appeal to practical considerations to justify the need to make explanations. Agents making inferences and agents using the produced conclusions have limitations.

One such explanation – appealing to practical considerations – that has been suggested [Dowe et al., 2011] for this need to explain is that, in real world settings, there exist restricted channels of communication. Since it is not always possible to communicate as much information as is needed, explanations are created to distil the essential information.

A similar explanation is that hypotheses which identify causes or select specific unob-

servable hypotheses are typically computationally easier to derive predictions from. In section 3.1, it will be argued that this is the most important motivation for explanations in formal/academic contexts.

1.14 The Principle of Multiple Explanations

The desire to not rule out any hypothesis consistent with one's observations is sometimes referred to as the Epicurus principle of multiple explanations. It may be stated as, *"Keep all hypotheses that are consistent with one's observations"*.

For prediction, following this principle might be ideal, but the principle seems to conflict with the common practice of inductive inference. Parameter estimation and model selection methods are almost always amplicative – violating the principle. This seems to imply that the Epicurus principle must somehow be incompatible with some essential aspect of inductive inference. Either that, or parameter estimation and model selection are bad forms of inductive inference.

To resolve this conflict we suggest that it is necessary that inductive inferences – the belief distorting type – be understood in the context of *practical limitations* and *limited practical investment*. The produced conclusions are not intended to be committed to indefinitely, nor to be used in all contexts.

If, for example, physicists in the past have concluded – based on experiment outcomes – that light is a wave and not a particle, that does not prevent later physicists from revising the conclusion, if contradicting evidence were to hypothetically mount against it. In the short-term, science seeks informative conclusions – conclusions which do rule out some hypotheses consistent with the observations.

In the short-term, new studies are often built on past conclusions which, while not being strictly certain, are considered probable enough. In the long-term, however, scientists tend to operate in a manner conforming more to the Epicurus principle: when contradicting evidence mounts against a theory, it will eventually be revised.

In philosophy of science, this process was studied by Kuhn [1962] who describes two types of science: evolutionary and revolutionary. Evolutionary science works within a paradigm and does not challenge the core assumptions of that paradigm. When enough

evidence has accumulated against those assumptions they are eventually revised and a new paradigm is born, that is revolutionary science.

We propose that theories explaining belief distorting inductive inference should not be limited to being concerned with the beliefs of ideal agents, they should be concerned with the process of forming beliefs intended for limited agents and limited applications. The inferences and resulting belief distortions that we will look at under the label interest-relative induction, should not be thought of as forming an indefinite chain of belief updates; that would lead to amplicative mistakes accumulating. Instead, inductive inferences might be imagined as temporary branches of a tree. The agent remembers some abbreviated version all observations (forming the trunk of the tree); temporary inductive inferences are created for specific tasks (forming branches in a nested structure); and these are eventually discarded.

1.15 Motivating Induction

In section 1.12 interest-relative induction (definition 5) was defined as belief distorting inference where the belief distortion (definition 4) is both *motivated* and *justified* by a *valid interest*. This section considers what it means for belief distortion to be motivated.

In section 1.14 it was argued that the use of parametrised models (which employ unobservable entities) and selecting hypotheses – as opposed to just using some joint probability distribution over all observables (a parameterless model) – is motivated by practical limitations. When the data source from which an inference is to be made conforms exactly to the model assumed a priori, belief distorting methods are suboptimal for decision making. If an estimator guesses the value of some unobserved proposition (is amplicative), an environment can be defined for which the outcome of an important decision depends on that proposition alone. If the estimator simplifies by forgetting some observed proposition (is destructive), again, an environment can be defined for which the outcome of an important decision depends on that proposition alone. If the estimator distorts its degree of belief in some proposition in any way that does not follow deductively from the prior belief and observed data alone, the same can be done. How then can the use of belief distorting methods – instead of purely deductive methods – be justified? Why not base all decisions on pure prediction?

We have argued that practical limitations must be the answer; distortion can only make sense when the inference is placed in a realistic context involving limited agents. This concept is now refined by dividing these limitations into simpler and less vague classes. We propose three classes of *practical constraints* that may motivate the use of belief distorting methods: constraints on making the inference, constrains on communicating the conclusion, and constraints on using the inferred conclusion.

To make the realistic context more concrete, it will be described as an interaction between two agents. That there is some *acting agent* which is to use the inferred conclusion in some environment. There is a *premise source* which will produce a premise. There is an *inferring agent* which receives the premise, translates it to a conclusion and communicates the conclusion to the acting agent. The inferring agent must be capable of mapping premises to conclusions for any of the premises that the given premise source may produce. The acting agent has an environment $\mathcal{E}$ for which a set of behaviours B is defined. The acting agent can receive a conclusion from the inference agent and will use it to choose a behaviour from set B. As mentioned in section 1.11, the word "premise" is used here to refer to a combination of observed data and a prior assumptions; so, a premise source produces both observations and a priori beliefs.

By placing the inference task in the context of these two agents, practical constraints have something to belong to. The three classes of constraints will now be defined.

Definition 6. *An* **acting constraint** *applies to an acting agent within some environment for which a set of behaviours is defined. It is any constraint restricting the agent's ability to compare the relative value of candidate behaviours under a given conclusion.*

Acting constraints arise because of limitations on the acting agent's mind: limitations on how it can translate belief into action. Acting constraints may exist because of time or memory constraints on the acting agent's mind. It could also simply be that the agent is programmed in a rigid way and can only make certain types of calculations or receive certain types of conclusions. As an example, imagine a program which can only use beliefs stated as if-then-rules (e.g., if x is true then y is true). When given a conclusion that cannot be written in that form, it will not be able to compare the value of behaviours from the set B. For all intents and purposes, it cannot hold that belief. As another example, imagine translating some concept for a five year old. You do not get to reprogram the child but must work within the existing constraints. If the translation

does not simplify enough then the child will not know what behaviours the conclusion entails.

Aside from acting constraints, there are also constrains on how inferences can be made. The inferring agent must be implemented physically and will have its own constraints.

Definition 7. *An* **inference constraint** *is any constraint on the inference agent's ability to map premises to conclusions for the given premise source.*

Finally, the inference agent must communicate the conclusion it chooses to the acting agent, any constraint on how that communication can be done will be called a communication constraint.

Definition 8. *A* **communication constraint** *is any constraint on how the inference agent can communicate conclusions to the acting agent.*

The combination of inference constraints, communication constraints and acting constraints will be referred to as *practical constraints* (definition 9).

Definition 9. *A* **practical constraint** *is the combination of all inference constraints, communication constraints and acting constraints that apply to a given inference task.*

The term *practical limitations* is used in this thesis to refer to the collection of all limitations applying to an inference task. Practical constrains are therefore a subset of practical limitations. We will not claim to have identified here all types of practical limitations that may exist. We propose that our concept of practical constraints captures most of what "usually validly" motivates belief distortion. Our defence for this position is that – as will be demonstrated in this thesis – models based on this principle can express (describe formally) a wide range of interests for any well defined data source. While there might not be a way to prove with finality that our practical constraints captures all possible valid motivations for belief distortion, the claim is falsifiable. Evidence against it may be provided by suggesting motivations for belief distortion which seem intuitively valid but which cannot be expressed as practical constraints.

1.16 Justifying Induction

In section 1.12 interest-relative induction (definition 5) was defined as belief distorting inference where the belief distortion is both motivated and justified by a valid interest. In the previous section motivation was discussed; in this section we consider what it means for a belief distortion to be justified.

Even if clear motivations for using induction can be given in terms of practical constraints (definition 9), that by itself does not justify using belief distorting induction. For justification there must also be some reason for believing that the inference can accomplish the desired goals despite distorting beliefs.

To justify the use of belief distortion, we propose, one thing is necessary: limited practical investment. No matter how an estimator defines the best estimate, if it distorts belief then it is possible to define some environment for which that estimator will be significantly worse than some other estimator. To justify belief distortion, one must accept that the conclusions will be appropriate for some environments but not for all possible environments.

As example, imagine that a belief $p(e) = 0.999$ is held about event e. Now imagine that because of some practical constraint, the event e must be estimated. A credence must be translated into an all-out belief. Assume that expected utility is used to measure the value of the estimate. Let $M = \mathcal{M}_{2,2}(\mathbb{R})$ be the set of all pay-off matrices. The expression $\mathcal{M}_{2,2}(\mathbb{R})$ denotes the set of all two-by-two real-valued matrices. A pay-off matrix defines four utility values: a reward for true positives, a reward for true negatives, a penalty for false positives, and a penalty for false negatives. Clearly, there is some $m \in M$ for which it will be better to estimate that e is false even though $p(e) = 0.999$ is believed – just set the penalty for false positives to a high enough value relative to the other penalties and rewards.

Let $N \subseteq M$ denote the set of pay-off matrices for which it is better (in expected utility) to estimate that e is false. If someone does choose to estimate that e is true, what does that imply about their belief about the intended environment? It could mean that they have ruled out that the intended environment may come from the set N. Alternatively it could mean that some weighting has been defined over the set M such that the expectation favours estimating that e is true. Note that if such a weighting is

present as a probability density q over the set M, then by integrating members of M weighted by q, a single new pay-off matrix can be obtained which is effectively equivalent.

By making an estimate, something is said – implicitly – about what the intended environment is and is not. Conversely, without saying something about what the intended environment is, nothing can be said about the expected utility of the two possible estimates. Compromise in belief distortion can only be resolved when investment of the acting agent is limited in some way.

1.17 Environments

In the previous section, it was proposed that for an inductive method to be justified it must be intended to be used in some, but not all, environments. We now look at how decision theory is used to represent such environments formally.

In decision theory, it can be imagined that some agent has a set of possible behaviours B to choose from. The outcome of behaviour $b \in B$ has expected utility $u(b, \omega)$ when true the state of the world is $\omega \in \Omega$, where the set of possible elementary world states is Ω. The utility function has the form $u : B \times \Omega \to \mathbb{R}$ and may hide many complicated interactions between aspects of the world and the decision. The elements of B need not be elementary decisions but could be complex programs (behaviours) instructing what to do in (possibly infinitely) many situations. Entire contingency plans can be represented as a single decision if one wishes to avoid defining elaborate environments with nested decisions.

Given a belief $p(\cdot)$ about the probabilities of the different possible states of the world being true, it can been argued that the best decision $b \in B$ is the one that maximises expected utility $u(b) = \sum_{\omega \in \Omega} p(\omega)u(b, \omega)$: this method will maximise utility if the decision is repeated indefinitely in the same context. Preferring expected utility, as opposed to some other quantity (e.g. worst case utility), is perhaps not the only valid path to investigate; for this thesis we restrict consideration to this measure.

Let Λ be the set of possible premises that the premise source may produce. Let Θ be the set of conclusions that are allowed by the practical constraint. Let $p_\sigma(\cdot)$ denote the probability distribution over the set of events E that is the meaning of premise $\sigma \in \Lambda$.

Similarly, let $p_\theta(\cdot)$ denote the meaning of conclusion $\theta \in \Theta$.

Imagine that the premise $\sigma \in \Lambda$ is not allowed by the practical constraint as a conclusion ($\sigma \notin \Theta$). It is now necessary to translate σ to a member of the conclusion language Θ. Through this process, belief may be distorted, $p_\theta \neq p_\sigma$.

Imagine that an inference has now been performed and the acting agent choose a behaviour based on p_θ. It chooses a behaviour $b \in B$ which maximises $\sum_{\omega\in\Omega} p_\theta(\omega)u(b,\omega)$ while maximising $\sum_{\omega\in\Omega} p_\sigma(\omega)u(b,\omega)$ would give the true optimal decision (under the assumption that the data source conforms to the meaning of premise p_σ). How good the translation from p_σ to p_θ is, depends here on the structure of the utility function u. This demonstrates that environments can help measure how good a belief translation is.

A version of this approach is used for the estimation method known as 'Bayesian estimators'. These methods select from a set of hypotheses Θ by minimising a loss function. Loss functions are effectively negative utility functions. A loss function $U(\hat{\theta}, \theta)$ for some parameter takes as arguments the estimated parameter $\hat{\theta}$ and the true parameter θ. The expected loss for some estimator $\delta : X \to \Theta$ (where X is the space of possible observations and a random variable of Ω) is then,

$$\int_\Theta [\ p(\theta) \sum_{x\in X} U(\delta(x), \theta)p(x|\theta)\ ]\ d\theta\ . \tag{1.4}$$

Loss functions abstract away the decision process, but U may be thought of as being derived from some decision based environment. As an example, imagine that Θ is the mean of a Gaussian distribution of known variance. Some samples from the distribution have been observed. One loss function one might choose is the squared error $(\hat{\theta} - \theta)^2$, where θ is the true mean and $\hat{\theta}$ is an estimate of it. The loss function abstracts away the decision process. In this example, the decision process is never explicitly considered in constructing the loss function. Loss functions like the squared error tend to be used because they make for simple mathematical solutions and are easily comprehended.

Note that with the loss function $(\hat{\theta} - \theta)^2$, if θ represents an unobservable then the utility is defined directly with respect to an unobservable. This would mean that for a given estimate $\hat{\theta}$, when given the true value of θ, the state of all observables – including future data – becomes irrelevant. This type of loss function cannot be derived from a real environment. If decisions in the environment can distinguish differences in the value

of θ directly – not using observables as intermediates – then the environment defines an experiment to observe θ; so θ must be observable. We will require that environments not be defined directly in terms of unobservables. Unobservables will be relevant to environments only through observables.

We will not be using loss functions for our model of interest-relative induction. Our approach to environments is to define them as a set of behaviours B and a utility function $u(B, \Omega)$. Our reason for using the more explicit environment definition, as opposed to using the loss functions of Bayesian estimators, is that they conflate agent and environment.

For agent and environment to be separated, it must be possible to imagine how the agent might be replaced with a different agent which has different limitations, while the environment remains the same. This separation is necessary in order to discuss the effects of practical constraints. Loss functions are defined in terms of the parameter to be estimated. Some other agent which estimates a different parameter, may be imagined in the same environment, but that would require that a different loss function be defined. The new loss function would need to have something in common with the old one, and so, must be derived from something that they have in common. Environments can be used to define that common foundation because they are defined in terms of behaviour instead of belief. This separation (of agents and environments) ensures that the limitations of specific agents – like the cost of having to compute actions from beliefs, or the inability to optimise actions from some beliefs – are not forced into the environment definition. There must be a separate formulation of agent limitations (practical constraints). For our purposes, the loss function approach is too restrictive.

1.18 Defining Interest

In section 1.12, interest-relative induction was defined (definition 5) as belief distorting inference where the belief distortion is both motivated and justified by a valid interest. In section 1.15, motivation was defined as practical constraints (definition 9) which are in-turn composed of inference constraints (definition 7), communication constraints (definition 8) and acting constraints (definition 6). In section 1.16 we proposed limited practical investment as the justification – the reason for believing that belief distortion

can be guided. It is now possible to propose a definition for interest.

Definition 10. *For an inference problem, an* **interest** *is composed of an environment and a practical constraint.*

Given such an interest, it is possible to guide belief distortion. Definition 10 captures both that which motivates the use of belief distorting induction (practical constraints, definition 9) and that which is needed to determine what the best conclusion is given the practical constraints (an environment, see section 1.17).

We will try to support this definition by demonstrating that models based on it (see chapter 2) can express a wide range of interests for any well defined data source and can produce useful and general insights into the relation between interest and induction.

1.19 Convergence vs Immediate Expectation

One commonly used way to defend an estimator is by appealing to convergence. If an estimator estimates parameter θ, it is desirable that as more data becomes available, the estimated value converges on the true value – either exactly or to within an arbitrary distance. Whether a given estimator has this property depends on the estimator used and also on the relation between the chosen model and the true data source.

While convergence is desirable, it may also be inadequate by itself when we know that only a certain amount of data will be available. A good example might be a program that is to predict results for the Olympic games. The 4 years gap between games and the continued evolution in the nature of the games means that little data might be relevant past two or three decades. Aside from that, it might be easier to design an algorithm that will be used for the next 12 years and after that to make a new one, as opposed to designing one that is intended to be used indefinitely.

As a measure of performance, convergence makes sense if there is a data-process and if the need for the inference method continues well into that process. For short-term applications, performance measures are needed that make sense for the actual data available and which say something about the current inference rather than some hypothetical process that might in reality not continue that far into the future.

Another approach is to look at how fast an estimator converges to the correct value – a property which is confusingly sometimes referred to as bias. This solution sadly still requires that a data process be assumed.

Bayesian estimators (as described in section 1.17) employ a third approach. The expected utility that is optimised can apply to the single inference that is currently considered. There is no need for a (possibly imaginary) data-process stretching infinitely into the future. This immediate expectation approach does however have a big weakness, its ability depends very strongly on the chosen prior belief. Under the assumption that the true data source is just as the prior assumes, the measure is ideal. If this assumption is incorrect, the measure may be very far off.

Such measures of immediate expected utility – measures which apply directly to the current inference – conflict with the philosophy behind objective priors. If the prior is chosen for its flexibility (e.g. an algorithmic prior), rather than the belief that it actually contains information about the data source, then there is less reason to take immediate expected utility seriously.

As we see it, the problem of deciding what should be the measure to optimise inductive algorithms for, is one that depends on whether a short-term or long-term application is intended. If goals are short-term, more immediate measures are needed and prior beliefs must be assumed to contain real knowledge; if they are long-term, prior beliefs may be chosen with long-term flexibility in mind. Convergence is meaningful for a data-process, immediate expectation is meaningful for a given inference.

While it is desirable to formulate our work on interest-relative induction without committing too much to subjective or objective priors, the incompatibility between short-term and long-term measures forces some commitment. Our work is concerned with belief distorting induction. It was argued in section 1.14 that belief distortion is acceptable for inferences with limited life-spans while long-term inferences best avoid distortion. For this reason, our model of interest-relative induction will be formulated using an immediate measure of inference value. Models that focus on the single next inference rather than a process of repeated inferences will be used.

1.20 Separation of Priors and Interests

When designing a probabilistic model to be used by an inference algorithm, the goal for a Bayesian subjectivist is that the model reflects what is believed about the true data source. As available data per parameter to be estimated decreases, the importance of good prior knowledge increases.

Other models, by contrast, are designed not to reflect what is known about a specific data source but to be capable of imitating a wide range of data sources and to minimise how much is assumed (objective priors and algorithmic priors were mentioned in section 1.11). The "objective" approach has its benefits when there is a lot of data and when very little is known about the data source. It is less good when little data is available or when the number of parameters to be estimated increases as the observed data increases.

Being true to what is believed about the actual data source and allowing for the model to emulate many possible systems, is not always the only consideration when designing a model. Interests are often also built into a model. An example will now be used to illustrate.

Imagine that for some set of individuals, for each individual, 3 symptoms are measured and it is known and recorded if they have some disease. For patient i we denote symptom $j \in \{1, 2, 3\}$ with $x_{i,j} \in \{0, 1\}$ and denote the presence of the disease with $y_i \in \{0, 1\}$.

It is also known that both age and gender influence the 3 symptoms. The age and gender attributes have not been measured – they are observable, but have not been observed. Figure 1.1 illustrates what is believed about the causal structure of the data source. Here, a node at the beginning of an arrow has a direct causal influence on the node that the arrow points to. A node is conditionally independent of all other nodes given all the nodes that have arrows pointing to it. For more on the relation between conditional independence and causation, and on these types of diagrams, see [Pearl, 2009].

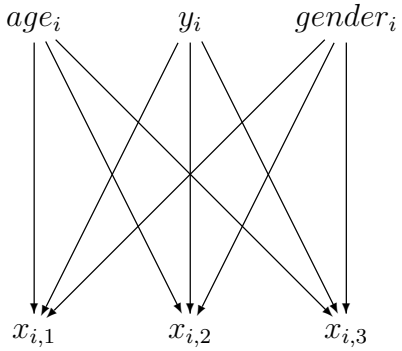


Figure 1.1: *A cross-section (for a single individual) of the ideal disease model. The top nodes represents the symptom causes while the bottom nodes represent the symptoms. The arrows represent causal and conditional independence relations.*

Now imagine that a model is designed which uses a conditional independence structure that mirrors this causal structure – which has been elicited from domain experts. Any algorithm that is meant to predict y given x for future individuals will have to make some inference about the relations between the 3 symptoms and the 2 unobserved attributes (age and gender). Such an algorithm would be computationally expensive: the possible permutations of the two unobserved attributes would have to be integrated over, or their effect somehow approximated.

An alternative approach is to simplify the model by excluding the unobserved attributes. Figure 1.2 shows the causal structure of this simplified model. This is the structure of the Naive Bayes classifier. Any algorithm that attempts to make predictions using this simplified structure is less ideal and the predictive performance – in theory – will be worse than a good algorithm that speculates intelligently about the influence of the unobserved attributes.

A third approach to this diagnosis problem is to reverse the assumed causal structure. Figure 1.3 shows the new structure of conditional independencies. This is a common solution when the relations between attributes, which are not to be predicted about, are complex and not of primary interest. For this prediction problem we are only interested in the conditional probability of y given x. Popular algorithms which use this structure include: decision trees, neural networks, and support vector machines. Such predictors/classifiers can often perform well even when they contradict what is believed

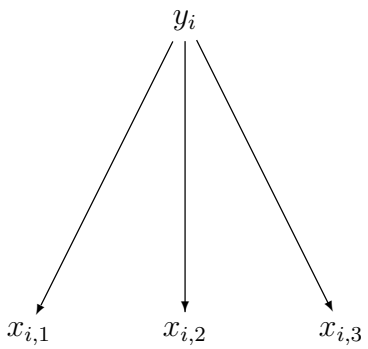


Figure 1.2: *A cross-section (for a single individual) of a simplified disease model. The top row has the unobserved symptom causes removed.*

about the true causal structure. This is because such algorithms are often designed to be capable of mimicking a wide range of conditional probability structures (y given x) and do not waste resources trying to explain the relations within x. The thesis [Jebara, 2001] is built on this principle.

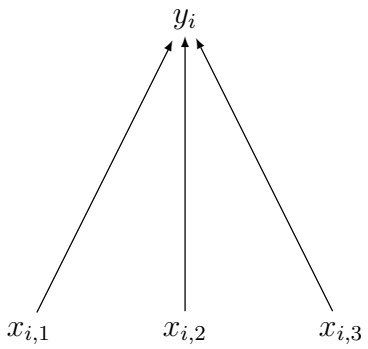


Figure 1.3: *A cross-section of the discriminative disease model for a single individual. The directions of causal and conditional independence relations have been reversed while the unobserved causes have been removed.*

Many models following the structure of figure 1.3 are part of a category known as *discriminative* models. Figure 1.1 is an example of a *generative* model as it represents the relations between all observations with a structure that reflects the nature of the assumed data source. The difference between generative and discriminative is explored in [Jebara, 2001] and is also discussed in [Dowe, 2008, sec. 0.2.5, p541]. Discriminative

models are good when classification of future items is what is of interest. Generative models are preferable when trying to explain the true data source.

The model for figure 1.1 was chosen to reflect what is believed about the true data source. The model for figure 1.3 was chosen to suit the goal of predicting y given x; it would be of no use for many other goals – such as predicting 2 symptoms given the other symptom and the disease. One model is based on a priori beliefs about the true data source and is not concerned with what the model will be used for. The other model is designed with a specific use in mind. We will use the terms *interest-free* and *interest-specific* to describe the difference between these models.

Assuming that the prior knowledge about a data source is good – or alternatively, appropriately flexible – interest-free models have more uses. They may require inference algorithms which are much harder to implement and which require more computation.

When trying to understand the role of interests in inductive inference, it is desirable to treat prior beliefs and interests as separate things. It seems to us reasonable to suggest that the language used to expresses prior belief about a system should be different from the language used to express interests in that system. This could be seen as a desire to separate normative beliefs (stating what is desired) and positive beliefs (stating how things are believed to actually be).

The ideal interest-free model can be imagined as the one that is believed to be appropriate for the largest range of interests. A good way to answer the question, is this model interest-free? is to imagine using the same model on the same data source for a different purpose.

It seems difficult to say if absolutely interest-free models can always be constructed; by limiting the variables included in the model to the system of interest, one is already saying something about what is of interest. It could be argued that an algorithmic prior should be considered absolutely interest-free as one can be constructed with no consideration of what the data source is.

Unfortunately, the concept of separating priors and interests is not one that is given much attention in the literature. We are not advocating that separation is always desirable. If only a specific interest is adopted then there are practical advantages to formulating an interest-specific model. For our goal of exploring the relation between

interest and inductive inference, it is desirable that they are formulated as separate things. For this thesis, it should be assumed that models formulating prior beliefs are designed to be as interest-free as possible.

The intention here is not to suggest that priors should never also include normative information. The intention is to argue that separation is more useful when analysing inductive principles. As an example, consider two agents: The first agent believes, "*It will probably rain later so we should not go to the beach.*" The second agent believes, "*It will probably rain later so we should go to the beach.*" These two beliefs are both normative and positive. Both can be useful beliefs for determining behaviour. If the goal is to compare the beliefs of these agents and how they lead to different actions, communications and inferences, it might help to be able to recognise that they agree on how the world is but disagree on how desirable it is to get wet.

1.21 Motivations for Interest-Absolutism

If a given inference method needs to be told about the specific interests that applies to the inference task for it to be able to produce a conclusion, it can be referred to as being interest-relative. Those methods which need very little information about the specific interest to make inferences will be called *interest-absolute*.

Amongst the Bayesian community, some inductive methods which are more interest-absolute have gained popularity. These may be estimators which use some standard loss function (like the squared-error loss function or the log-loss function) or methods like minimum message length (MML) which define the best estimate without any explicit reference to utility. The term *standard loss function* is used here to refer to those loss function which are not tailored to each application, instead, they are loss functions designed to lead to inferences which are good for a wide range of environments and data sources. These methods are objective in the sense that they do not require specific interests to be specified in much detail for new problems; but, they could still be seen as having implicit interests.

So far it has been argued that belief distorting induction is motivated by practical constraints (section 1.15); and that only limited environments of interest can justify the use of belief distorting inductive methods (sections 1.16 and 1.17).

All this begs the question, why are inductive methods which attempt to be interest-absolute developed and desirable? Even when the goal is for an inferred conclusion to be useful in all environments, if there is a practical constraint preventing a deductive inference method from being applicable, then surely some information – even an educated guess – about the intended environment would be preferable to nothing. Indeed, the more precisely the environment is specified, the better the inference method can know how to compromise when selecting its conclusion.

We can think of four motivations behind the popularity of interest-absolute estimators: the first is *ease-of-use*, the second is *predictability*, the third is a *desire for objectivity*, and the fourth is *re-usability* of the inference.

If an inference is to be made, a loss function could be defined and the estimator solved for it, but this would be time consuming and require mathematical skill. If an existing solution exists which roughly fits one's purposes, that estimator may be used. This desire for *ease-of-use* is one reason why developing interest-absolute methods is important.

Imagine you were to define your own loss function for some problem and then to solve an estimator for it. You may then run into unexpected properties of the estimator which must then be explored from scratch. Arguably these undesired properties might be the result of not defining a loss function that exactly captures your goals, or it may stem from some shortcoming of the estimator's search algorithm. For an estimator which has been used by many others and which is intended for more general interests, there may be proofs and evidence available giving some guarantees about the estimator's properties. This *predictability* of interest-absolute methods is one important motivation for their development.

One concern that scientists may have with specifying utilities when making inferences is that it makes results appear subjective. Scientists tend to be concerned with finding "the truth" rather than some subjective truth. To use a method that is very clearly dependant on subjective choices opens one's findings to criticism. Those who use Bayesian methods with subjective priors have already given up on objectivity to a degree, however, subjectivity of priors is a separate issue from relativity to interests. One who accepts that inductive inference is relative to prior beliefs does not necessarily accept that inductive inference is relative to interests. This *desire for objectivity*, we

suspect, has contributed to the development of interest-absolute inductive methods.

Sometimes beliefs are formed without any particular or exact goal in mind. Science is sometimes seen as being concerned with finding truth free of any particular application. This process might be thought of as looking for hypotheses which will be useful for many applications. For science, it is not generally known, before the inference is made, what all the uses for the inferred conclusion will be. For this reason it may be important to select hypotheses which are *re-usable*, useful for many different specific interests. It is also desirable that theories about the world do not change too often. Agents have to allocate resources to update a belief and will not necessarily update beliefs after every new observation.

If interest-absolutism is desirable, how can that be reconciled with the argument presented that belief distortion is necessary under practical limitations and that only limited investment can justify it? One way to achieve some degree of interest-absolutism for inductive methods is to create broad classes of interests. More general interests might be composed by putting together many specific interests. It would then be the severity of the practical constraint that limits how general an environment one can choose to optimise for. The limitations of such an approach are explored more carefully in sections 2.12 and 2.13. There it can be seen that for a given practical constraint, it is possible to say that some environments are strictly more invested (are concerned with the same and more entities/propositions) than others; but, a fully invested environment can only be approached when there are no practical constraints.

1.22 Belief Attribution

In section 1.9, our motivations for introducing a new approach to defining belief and how it relates to action were presented. We use a class of beliefs which we refer to as *incomplete Bayesian beliefs* – or just *incomplete beliefs*. This class is similar to Bayesian beliefs but allows for more vague beliefs to be held. We also define the relation between beliefs and actions for limited agents explicitly in terms of acting constraints (definition 6). This section describes the method that is used, technical definitions are presented in sections 2.3 and 2.5.

Our incomplete beliefs use what is known as imprecise probabilities. An imprecise probability is a set of probability functions. As an example, let p be some proposition; the belief $\Pr(p) = 0.7$ is a precise probability, the belief $\Pr(p) > 0.5$ is an imprecise probability – it can be represented as the set of all probability functions conforming to that inequality. We use the term "incomplete beliefs" to refer to our method for relating imprecise probabilities to hypothetical behaviour. For other approaches to relating imprecise probabilities to dispositions to behaviour, see [Walley, 2000]. Our method is arguably more general, though, it is built on the assumption that agents always act to maximise expected utility.

Let E be the set of observable events with Ω being the corresponding elementary event set. Let Φ denote the set of all possible incomplete beliefs concerning E. Let $F = \Omega \to \mathbb{R}$ be the set of all functions mapping all elementary world states to utility values. Imagine now that there is some *ideal agent* which has no practical limitations – it knows the implications of any belief from Φ exactly. This agent will be denoted by $\mathbb{A}$. Imagine that $\mathbb{A}$ holds incomplete belief $\phi \in \Phi$. Let $f_1, f_2 \in F$ represent the rewards for two possible choices of some hypothetical decision. We now ask $\mathbb{A}$ which option has the best expected utility outcome under belief ϕ. It must reply with one of four possible answers. Option 1, f_1 is better. Option 2, f_2 is better. Option 3, f_1 and f_2 are equal. Option 4, ϕ is too vague to entail the answer. Such incomplete behavioural preferences are also explored in [Bales et al., 2014].

The meaning of ϕ is now captured by the complete collection of $\mathbb{A}$'s answers to all pairs in the set F. With this method it is possible to define beliefs which may be vague. If the agent does not answer with option 4 for any pair from F under some $\phi \in \Phi$ then the belief ϕ is not vague and is equivalent to a complete Bayesian belief.

The exact definition of incomplete beliefs is given in section 2.3 – some mathematical considerations are omitted here for brevity. This is a very flexible language of belief. For example: For $e \in E$, the belief "$\Pr(e) > 0.5$" is not a complete belief but it is a member of Φ. Any conditional probability that can be defined within E is a member of Φ. For any random variable of E there will be members of Φ which only say things about that that variable.

Next, the method by which incomplete beliefs are related to behaviour for limited agents is described. The idea that rationality can be defined for limited agents according

to how near optimal their behaviour is within their limitations is not new (see [Baron, 2005]). We introduce a method for defining shared meaning among limited agents – having possibly different limitations – which is necessary for considering belief distortion justification.

Each limited agent i will be assumed to have a corresponding environment $\mathcal{E}_i$ with behaviour set B_i and utility function $u_i : \Omega \times B_i \to [\,\check{i},\, \hat{i}\,]$ where $\check{i}, \hat{i} \in \mathbb{R}$ are lower and upper bounds in utility respectively.

If limited agent i holds belief ϕ and it is known that i does not always behave optimally in its environment under its held belief, it is not possible to define the meaning of ϕ using i's behaviour – as was done for the ideal agent $\mathbb{A}$. If the assumption of optimal behaviour is dropped the relation between behaviour and action degrades and the "meaning" of the belief becomes dubious. We will get around this problem by assuming that there is some community of agents $\mathcal{A}$ sharing a common language about world E. The ideal agent $\mathbb{A} \in \mathcal{A}$ is assumed to be the one agent that can act optimally in all environments. By pretending that there is some ideal agent $\mathbb{A}$, the intended meanings of beliefs from Φ are retained through the shared language. For each $\phi \in \Phi$ there is some code in the common agent language. That code can be sent to any agent forcing it to adopt the corresponding belief in whatever form it is represented in its mind. If $i \in \mathcal{A}$ is given the code for $\phi \in \Phi$ and adopts a different behaviour than $\mathbb{A}$ would have, we assume that i chose incorrectly because of its acting constraint (definition 6); and not because it misinterpreted the meaning of the code for ϕ. It chose incorrectly because it cannot compare the relative value of candidate behaviours under the given conclusion. With this method, a relation can be defined between belief meaning, behaviour and acting constraints. The acting constraint of i can be determined by looking at how its behaviour deviates from that of $\mathbb{A}$ when given the same code.

There could be many ways to formulate acting constraints. We will not claim to have found "the" best way to do it. Instead, a method is now presented which we have found to balance manageability and generality. To understand the reasoning behind our method a hypothetical agent implementation might be considered which computes its behaviour choice within computational limitations using a local search algorithm. It uses the following algorithm:

step 1: agent i receives conclusion ϕ

step 2: some candidate behaviour $b \in B_i$ is produced

step 3: the value of b under ϕ is calculated

step 4: the algorithm either times-out or returns to step 2

step 5: the best behaviour considered is selected

The result of following this pseudo-algorithm is that only some subset of B_i is considered. That set is determined only by the the given conclusion ϕ and the implementation of agent i. The acting constraint of agent i can therefore be defined as a function U_i of the form $\Phi \to \mathcal{P}(B_i)$ where $\mathcal{P}(B_i)$ is the power set of B_i. When agent i receives incomplete belief ϕ it must choose a behaviour from $U_i(\phi) \subseteq B_i$ which is optimal for ϕ when restricted to $U_i(\phi)$ according to how $\mathbb{A}$ would interpret ϕ. When $U_i(\phi) = \emptyset$, there is no behaviour that i can choose. Here it will be said that i cannot hold belief ϕ – ϕ is not in its *conclusion language*.

There are many special cases that have not been mentioned here. Some beliefs from Φ will be too vague to define what the best behaviour is for some environments. The best behaviour might exist only as an asymptote. Not all functions from $\Phi \to \mathcal{P}(B_i)$ are valid acting constraints. These and other issues will be dealt with in chapter 2.

With this method of belief attribution, the link between belief meaning and behaviour of limited agents is preserved. The less U_i restricts what behaviour subset can be considered, the closer agent i's behaviour will get to that of $\mathbb{A}$. In section 2.11, different classes of acting constraints will be defined according to the structure of U_i. That allows classes of acting constraints to be related to types of belief distortion.

In [Holton, 2013] an argument is made against credences and Bayesian beliefs as models of real world agent beliefs. The argument there is that real world agents have limitations and as a result tend to only work with a small set of explicitly held all-out beliefs (running beliefs) at any time, and that the Bayesian method of deriving belief from hypothetical betting behaviour fails to capture this – it leads to beliefs with too many live possibilities. A "live possibility" is one which the agent is willing to consider during decision making. For example, I know that there is a non-zero probability that

an asteroid will destroy all life on earth next week but do not include that possibility in my reasoning for the future in any way. Holton [2013] argues that intention is necessary for belief to be defined for realistic agents. The reasoning there mirrors that of this thesis where practical investment (as we call it) is similar to "intention".

Where we disagree with Holton [2013] is in the assertion that Bayesianism and the use of credences is incompatible with agent limitations and the need for intention in defining beliefs. While a proper refutation of that position is beyond the scope of this thesis, we hope to demonstrate (using the method described in this section) that when acting constraints, agent investment, and a shared agent language are properly defined, Bayesian probabilities are not ruled out and are still useful. Sections 2.3 through to 2.6 demonstrate more precisely how this can be accomplished.

1.23 A Theory of Interest-Relative Induction

A definition for interest-relative induction has been given and some concepts important to it have been described. Interest-relative induction is defined as belief distorting inference where the belief distortion is both motivated and justified by a valid interest (definition 5). A valid interest is composed of an environment and a practical constraint (definition 10). Practical constraints, in turn, are composed of an inference constraint (definition 7), a communication constraint (definition 8) and an acting constraint (definition 6).

Interest can be understood by placing the inference task in the context of two agents: an *inference agent* and an *acting agent*. The inference agent will receive a premise from some set of possible premises. For the given premise, it must produce a conclusion and communicate it to the acting agent. The acting agent, not knowing the premise, will use the conclusion to act in its environment. The value of the translated belief is measured by considering the acting agent behaving in its environment in a way that is optimal for the inferred conclusion, under the assumption that the true data source actually matches the premise. The acting constraint captures the limitations of the acting agent. The inference constraint captures the limitations of the inference agent. The communication constraint captures limitations of the process by which conclusions can be communicated from inference agent to acting agent.

This scenario forms the basis of our theory of interest-relative induction, but it is still not specified clearly enough. To move forward, an unambiguous formal model will be created to describe the relation between interest and inductive inference. This model is developed in chapter 2. Some requirements that it should meet are proposed here.

Model Desiderata: the model must:

- be applicable to a large range of hypothetical data sources
- be capable of expressing a large range of practical constraints
- represent how practical constraints force belief distortion
- be capable of expressing a large range of environments
- represent how environments guide belief distortion

The model will be used to elicit generalisations about the nature of the relation between interest and induction. Models are idealised imitations of real phenomena in which all variables and relations are fully instantiated. A model cannot prove the assumptions on which it is built, it can only make more transparent the implications of those assumptions. The assumptions that will go into this model follow.

Primary Assumptions:

A) Prior belief is needed for inductive inference.

B) All observations have discrete outcomes.

C) The value and meaning of a belief is derived only from how it influences the actions of agents in environments.

D) Only practical limitations can be used to defend the use of belief distorting inference methods.

E) With the practical limitations taken into consideration, only the environment in which the conclusion is to be used can be used to defend preference between belief distorting inference methods.

The prior belief from assumption A should be interpreted either as prior belief in the Bayesian sense or as weightings over hypotheses as is used with algorithmic probabilities. Assumption B is made because measurements are always made to finite precision, the alternative would be to allow for an infinite amount of information to be observed. Assumption C fits roughly with how subjective probabilities (Bayesian probabilities) work and falls into the interpretation of belief known as behaviourism. Assumption D was defended in sections 1.13, 1.9, 1.14 and 1.15. Assumption E was defended in sections 1.14, 1.16 and 1.17.

The primary assumptions listed above are those which are considered essential to our interpretation of interest-relative induction. Below, some secondary assumptions are listed. These were needed to construct a complete formal model of interest-relative induction; they could be altered to produce essentially similar models. For example: some of our earlier models (not presented in this thesis) did not have communication constraints, some used orderings of outcomes instead of utilities, and some attempted

to avoid using worst-case utility measures. The secondary assumptions presented here are the best combination that we could make useful.

Secondary Assumptions:

F) All valid practical limitations belong to one of the thee following classes: inference constraints, communication constraints, and acting constraints.

G) For an acting agent, the value of an outcome should be defined by assigning a real number utility to it.

H) The value of a given inference should be measured using the expected utility of acting optimally for the conclusion under the assumption that the premise matches the true data source exactly.

I) When multiple optimal behaviours exist for a given conclusion where some have different values under the premise, a worst-case expected utility measure should be used.

Assumption F was described in section 1.15. There might be better alternatives for defining practical constraints; it is hoped that ours captures what should be considered "valid" motivation for belief distortion. Assumption G is made simply to make the model in chapter 2 possible. Assumption H is defended somewhat in sections 1.17 and 1.19. This assumption should make it clear that our model focuses on belief translation: the premise to be translated is not doubted. Assumption I is made to make the model in chapter 2 possible, as will be seen in section 2.6.

It is important to consider how this interest-relative theory of induction relates to deception and delusion. Assumption D says that belief distortion can be justified only by practical limitations but it does not say what practical limitations are not allowed. There is nothing here (or in our model) that explicitly excludes pathological practical limitations. One might then criticize this approach by saying that it allows for poor forms of reasoning to be justified. As an example, consider wishful thinking. Imagine that Gilgamesh cannot tolerate the idea that he will one day die. If he were to believe that death is inevitable he would freeze up and be unable peruse any goal other than achieving immortality (which is assumed to be futile for the sake of this example). This is an acting constraint as it restricts behavior under certain beliefs. Under our approach,

this constraint is allowed as a valid justification for belief distortion. Imagine that the belief, "Gilgamesh is already immortal", allows him behaviors that lead to greater expected utility than any other alternative. For him, believing this statement might be the most practical option. On the surface, it might seem like our approach could be used to justify wishful thinking here, but that is not the case. If Gilgamesh justified this conclusion by saying that it is what he wants the truth to be then that would be wishful thinking on his part. Our model is not concerned with how acting agent Gilgamesh justifies his beliefs, it is concerned with how some inference agent (possibly someone else or some "deeper" part of his mind) justifies the belief for him. The justification for the conclusion, "Gilgamesh is already immortal", under our approach, is that, given his severe acting constraint, this is the belief that will lead him to behavior that maximises expected utility. The fact that wishful thinking on his part also leads to this best conclusion is coincidence.

Our approach can justify seemingly delusional conclusions, but it only does so in the context of practical limitations. In the example above, the problem is that Gilgamesh has a pathological acting constraint. If, for some less constrained agent, the proposition, "Gilgamesh is not immortal", is preferable, we might label it as a 'less distorted' conclusion because the practical constraint behind it is less severe. The less distorted conclusion is not 'better' than the more distorted conclusion, it is *better for* less constrained agents. All real world agents are limited in some way. Absolutely undistorted conclusions may often have little practical value by themselves. The model we introduce in chapter 2 looks at how different practical constraints may be classified and compared to each other. It looks at how the properties of practical constraints may lead to different types of belief distortion.

It is desired that justified belief distortion – as represented by the model – does not include deception. The assumption of our approach is that the inference agent selects a conclusion that attempts to maximize expected utility for the acting agent. Whether this means that the inference agent 'deceives' the acting agent, or is just partially digesting the truth on its behalf, is a consideration that we place outside the scope of this thesis. Care should be taken to not apply these ideas too naively to human communication. One big problem that stands in the way is that in reality, it is very difficult for one person (inference agent) to sufficiently know the acting constraint of another individual

(acting agent) or to consider all situations in which a distorted conclusion might be used.

In chapter 2, the model will be developed and some of its properties revealed. Chapter 3 discusses the ideas developed in relation to select topics. In chapter 4, an interest-relative approach to inductive inference is proposed.

The model developed in chapter 2 should not be interpreted as describing an inference method. We do not suggest that algorithms realising it could ever be practical. It is a theoretical tool meant for eliciting principles which can in turn aid the process of designing inference algorithms.

Before presenting the model, this chapter closes by looking at some existing related work (section 1.24). The goal of these summaries is to note where they overlap with the topic of interest-relative induction, to acknowledge where we took inspiration from, and to mention where we deviate and why. Reading these summaries is not necessary for following the remainder of this thesis, they are simply included for completeness. Impatient readers are advised to skip forward to chapter 2 – or to section 2.19 if there is a desire to avoid the technical details of the model.

1.24 More Related Work

1.24.1 Model Theory

Model theory is concerned with the relation between models and theory. Theory is intended to apply directly to some area of the real world; it involves rules which are not too dependant on a specific formulation and as a result may be sometimes vague. Models define all entities and rules clearly, but possibly inaccurately. Models are simplified, internally consistent and complete instantiations of some aspect of the real world. Models are analogies to the real world and can help elicit or refute theory by making clear the implications of the assumptions that they are built on.

Giere [1999, 2004] explores this relation between models and theory. The process of simplifying some area of reality into a model is described as discarding unimportant details and distorting what can be distorted safely. This distortion of information is guided by constraints on the desired model form and on the process of creating the model. Different models of the same phenomenon may be constructed for different purposes.

Model theory, as described by Giere [1999, 2004], has had a strong influence on how we have approached the problem of interest-relative induction. The main difference is that model theory is concerned with the process by which models are constructed for scientific purposes. Our work is concerned with the process of inference after a model has been constructed.

1.24.2 Kuhn's No Unique Algorithm Theorem

Kuhn [1962] describes different desirable properties that hypotheses may possess. These include simplicity, accuracy and scope. Different scientists or communities of scientists may value these properties to different degrees. When trying to choose the best hypothesis from a given set of hypotheses, they will not all be able to agree, because of these differing values. As Kuhn sees it, there is often no unique "correct" way to weigh these desiderata against each other.

The gist of this theorem is the same as what motivates our work. Our work is intended to develop a method for examining the problem more precisely. We do not look at specific desiderata like simplicity, accuracy or scope. Instead, we use environments to define the most general range of possible desiderata. We also make explicit the difference between environments and practical constraints.

1.24.3 Knowledge and Practical Interests

Epistemology is the study of knowledge. One early example of a definition for 'knowledge' is that any statement which is true, justified and believed is knowledge. This definition has problems and alternatives have been proposed. Stanley [2005] argues that what constitutes knowledge is interest-relative. In fact, use if the term "interest-relative" in our work follows from its use in epistemology. Here it is argued that some belief is knowledge, based, not only on how it is inferred, but also on facts about the individual's environment. The examples used in section 1.1 to illustrate what interest-relative reasoning is were based loosely on examples from that thesis.

While it is suggested there that whether a belief is knowledge, depends on the probability of its truth combined with what is at stake in believing it, that work does not explicitly deal with probabilities, utilities or formal models. An epistemological approach was taken; making it very difficult to relate directly to our work, despite the strong similarities in the basic premise.

1.24.4 Pure Prediction and Decision Theory

Bayesian prediction simply means that one uses the posterior of the event of interest, given the observation, as the prediction for that event. If e is the event of interest; x is the observation; and p the prior, tying e and x together; then $p(e|x)$ is the prediction for e. Solomonoff induction [Solomonoff, 1964, 1996] can be thought of as a form of Bayesian prediction.

Decision theory is concerned with choosing actions based on credences to perform optimally in some environment – to optimise some utility function. Combining pure prediction with decision theory produces a method for choosing actions that is optimal

if the true data source conforms to the prior beliefs.

In [Hutter, 2005, Legg, 2008], decision theory is combined with Solomonoff prediction to define "super-intelligence" (the ideal intelligent agent). The AIXI agent, from that work, will eventually learn to perform optimally under certain assumptions about the data source and environment; thus, the combination of pure prediction and decision theory can – to some extent – even overcome the need for assuming that the prior contains information about the data source. This same approach is used by Hernández-Orallo and Dowe [2010] to define objective measures of intelligence applicable to different types of intelligent agents such as humans, animals and machines.

Performing prediction and decision making as two separate phases – one following the other without conflation – would remove the need for interest-relative induction. For a discussion about the absolute limits of prediction and intelligence, such a separation is appropriate. The separation does, however, not reflect how everyday reasoning, scientific reasoning or automated reasoning is actually done. We hold that it is more realistic to consider agents which make and commit, temporarily, to amplicative hypotheses.

1.24.5 Bayesian Estimators

Bayesian estimators were described in section 1.17. They are similar to the model that we will use for interest-relative induction, but differ in some crucial ways. The loss functions of Bayesian estimators do not represent practical constraints (definition 9) separately from environments, it conflates them (see section 1.17). Bayesian estimators also force the environment to be defined in terms of the parameter being estimated and rule out conclusions which contradict the premise.

Bayesian estimators can be defined for a wide range of interests and data sources, but we have not found them adequate for defining and analysing the concept of interest-relative induction itself.

1.24.6 To Predict or to Explain

Shmueli [2010] identifies two different types of interest: explanation and prediction. The common assumption that good predictive performance naturally leads to good explana-

tions is challenged there. Description is also identified as a type of interest. Shmueli points to a void in the statistics literature around these issues and the fact that introductory statistics text books tend not to treat prediction as a valid scientific endeavour – a situation, which we note, seems to be reversed in the machine-learning literature where prediction is often assumed to be the only meaningful measure of performance.

The approach employed by that work is to show that different modelling and inference methods are indeed commonly used by practitioners for prediction and explanation. We see [Shmueli, 2010] as evidence that a more comprehensive understanding of the role of interest in inductive inference is needed. Our work is intended to go beyond identifying a small list of interest types. It attempts to delimit the range of all possible interests. Our approach is less empirical and more concerned with developing a theoretical foundation for this topic.

1.24.7 Discriminative and Generative Learning

Where Shmueli [2010] identified three types of interest from the statistics perspective, Jebara [2001] identified two common approaches to machine learning: discriminative and generative. Discriminative learning is analogous to prediction. Generative learning is analogous to explanation, it is intended to say something about the mechanisms behind the data source.

Examples of discriminative inference algorithms include: decision trees, artificial neural networks, and support vector machines. Examples of generative methods include: Bayesian networks and unsupervised classifiers. Discriminative methods are good for prediction while generative methods are good for identifying causes and developing explanations.

The goal of [Jebara, 2001] was less to explore two types of interest, and more to develop hybrid machine learning algorithms that share some of the advantages of both approaches. One might say that they adopt a hybrid interest in doing so.

We believe that it is important to identify distinct interests that are already present 'in practise'; but, Jebara's goal of hybridising discriminative and generative interests suggests that it would be useful to not look at all interests as belonging fully to either one class or another, but to define a flexible language for describing interests in general.

1.24.8 MML and MDL

The minimum message length (MML) [Wallace and Boulton, 1968, Wallace and Freeman, 1987, Wallace, 2005] and minimum description length (MDL) [Rissanen, 1978, Grünwald, 2007] principles are used in practice to guide the design of inductive inference algorithms – often with the aim of producing explanations. These two are arguably the most widely applicable principles for that purpose. Section 3.5 will look at MML in more detail in relation to interest-relative induction. Some suggestions will be made for how MML may be applied using a more interest-relative approach. In chapter 4 those suggestions are demonstrated.

MML is a Bayesian estimation method which defines what the best estimate of some parameter Θ is, given: an observation $x \in X$, a prior distribution over the parameter $h(\Theta)$, and a likelihood function $f(X|\Theta)$. The MML literature does not explicitly mention interest much. In section 3.5 it is argued that there are several ways that interest information may enter MML problem specifications and solutions. It is also argued that this is a desirable property.

MDL differs from MML in small ways in theory. The MDL literature does deal with interest more explicitly but only in a limited way. A small set of possible purposes are identified and a separate version of MDL is advocated for each. These are parameter estimation, model selection and prediction [Grünwald, 2007]. To do model selection in MML, one (simply) integrates (or sums) out parameters – see, e.g., [Dowe, 2008, footnote 152]. MML is always an estimation method so there is no purely predictive version as with MDL. For more detail on the differences and between MDL and MML see [Wallace and Dowe, 1999, sec. 6.2], [Wallace, 2005, sec. 10.2] and [Dowe, 2011, sec. 6.7].

The approach of MDL makes sense when the interest matches closely enough either estimation or prediction, but when it doesn't, one may feel forced to transform the true interest into something that it is not. The approach of MDL to interest does not separate practical constraints and environments or properly identify their roles.

Sometimes, an inference is made for estimation, and prediction, and model selection, all at the same time. Sometimes these categories alone can not express specifically enough the actual interest. The solution we propose is not to try and list all the specific interests that should be considered; nor do we think that it is possible to solve the

difficulties in making these methods sufficiently interest-relative by defining more and more variations of them. Instead, we propose that the solution is to elicit general principles of interest-relative induction which can then be used to aid with decisions that arise when designing inference algorithms.

Chapter 2

A Formal Model of Interest-Relative Induction

2.1 Chapter Overview

In this chapter a model for interest-relative induction is presented. It is not suggesting that algorithms realising it could ever be practical. The model is intended as a theoretical tool for eliciting principles which can aid the process of designing inference algorithms.

Section 2.3 defines more precisely what we called the space of incomplete beliefs (from section 1.22). These will be used as beliefs for limited agents – agents which might hold beliefs too simple or incomplete to be captured by normal Bayesian beliefs. Section 2.4 lists some notation to be used in the rest of this chapter. Sections 2.5 through to 2.10 focus mostly on developing and presenting the model. Sections 2.11 through to 2.18 explore the properties of the model. Section 2.19 gives a summary of the conclusions of this chapter. These conclusions relate to the model itself, their wider implications are discussed in chapter 3. Readers wishing to avoid the technical parts of this thesis may skip to the chapter summary (section 2.19).

2.2 Partial Orderings

A brief introduction to partial orderings must be presented. A partial ordering over a set G is a relation $\geq$ defined over that set. It must hold the following properties for all members $x, y, z \in G$:

- reflexivity: $x \geq x$
- antisymmetry: $(x \geq y \wedge y \geq x) \Rightarrow x = y$
- transitivity: $(x \geq y \wedge y \geq z) \Rightarrow x \geq z$

Here $x > y$ is true iff $x \geq y$ is true while $y \geq x$ is not. The word 'iff' is used to mean 'if and only if'. Note that for a partial ordering there will be pairs $x, y \in G$ such that neither $x \geq y$ nor $y \geq x$ will be true – implying also that $x \not\equiv y$. Here, "$\equiv$" is used to indicate equivalence under the ordering, not to be confused with equivalence within the set G.

All readers should be familiar with total orderings. Total orderings have an additional requirement, either $x \geq y$ or $y \geq x$ must be true. An ordering over numbers is a simple example of a total ordering. For any two real numbers $x, y \in \mathbb{R}$, either one is greater than the other or they are equal. Partial orderings differ in that there may be pairs $x, y \in G$ for which none of the possibilities $x > y$, or $x < y$ or $x \equiv y$ are true. In that case, the pair is said to be incomparable. For comparable pairs, exactly one of those three options will be true. To summarise, partial orderings differ from the better known total orderings in that there can exist incomparable pairs under them.

The *greatest* element $x \in G$ is the element for which $\forall y \in G,\ g \geq y$. Similarly, the *least* element is defined as the $x \in G$ for which $\forall y \in G,\ g \leq y$. There might not always exist a greatest element. This could be because there are incomparable pairs or because the set G always has a greater member (for example the natural numbers has no greatest member, and the open interval $(0, 1)$ likewise has no greatest element). When they exist, greatest and least elements are unique.

An element $x \in G$ is *maximal* iff $\neg\exists y \in G,\ x < y$. *Minimal* elements are defined similarly. There could be multiple, non-equivalent, maximal and minimal elements for partial orderings; or there could be none (as with natural numbers).

2.3 Incomplete Bayesian Beliefs

In chapter 1 it was argued that the relation between inductive inference and interest could be modelled by imagining two agents, an inference agent and an acting agent. The inference agent receives a premise from some premise source, it then distorts the given premise to produce a conclusion that can be used by the acting agent. It is desirable that the model be general enough to describe a wide range of agents. A flexible language of agent beliefs is needed. In section 1.22 the concept of *incomplete beliefs* was described. This is a type of *imprecise belief*. Imprecise beliefs use sets of probability functions as beliefs. The term "incomplete beliefs" is used to refer to our method of relating imprecise beliefs to behaviour. This section covers the more technical considerations needed to define incomplete beliefs more carefully.

Ramsey [1931] defined subjective probabilities (including what we now call Bayesian probabilities) by imagining an agent being presented with betting decisions of different odds. The degrees of belief in different events may then be deduced from the those betting preferences. Incomplete beliefs are defined using the same approach but, for the considered agents, preference may be undefined for some decisions.

The observable world is delimited using a space of events E that is built from a set of elementary events Ω. Let P denote the set of all probability functions that can be defined over event set E. P will be called the *base belief set*. For $e \in E$ the probability of event e according to complete belief $p \in P$ is $p(e)$. The set of incomplete beliefs for P is the power set of P, it will be denoted by $\Phi = \mathcal{P}(P)$.

Let $\mathbb{A}$ denote the ideal agent (as in section 1.22). This is the agent that has no practical constraints. The meanings of members of Φ are all derived from this agent's hypothetical behaviours when holding Φ. Imagine that $\mathbb{A}$ is presented with a decision that has several options. Let f_i be the utility function defined over Ω for option i; it has the form $\Omega \to \mathbb{R}$. As a notational short-cut, square brackets will be used to denote expectation. For $p \in P$ the expectation of function f_i is denoted by $p[f_i]$.

The agent's preferences can be expressed as a partial ordering. Let $\phi \in \Phi$ be the incomplete belief that the agent holds. We define $f_i \geq f_j$ as true iff $\forall p \in \phi,\ p[f_i] \geq p[f_j]$. It can be shown that this defines a partial ordering. If $f_i \geq f_j$ then the agent prefers option i over option j. If $f_i \equiv f_j$ (equivalent according to the ordering) then the agent

has equal preference. If the pair is incomparable – i.e., $\neg(f_i \geq f_j) \wedge \neg(f_j \geq f_i)$ – then there is no preference defined for the agent. Incomparable pairs should not be confused with equal preference, they occur because ϕ is too vague to entail a preference.

If, for some agent, one knows for a large range of such decisions what its preference is, the largest conforming member of Φ can be attributed to it. If there is no subset of the base belief set P meeting this requirement then the agent's behaviour is inconsistent with belief space Φ.

The space of incomplete beliefs allows for a wide range of vague beliefs to be expressed. Some examples are now given. Let ϕ be the belief that event $e \in E$ is more probable than its complement. This can be defined as $\phi = \{\, p \in P \mid p(e) > 0.5 \,\}$. Notice that ϕ says nothing about other events which do not overlap with e. Let X be some random variable. Let ψ be the belief that the members of X are equally probable if event e is true. This can be written as $\psi = \{\, p \in P \mid \forall x, y \in X,\ p(x|e) = p(y|e) \,\}$. If X is not finite, ψ will be the empty set (since $\sum_{x \in X} p(x|e) = 1$ must hold), which would mean that there is no belief meeting the requirement. The beliefs ϕ and ψ can be combined using intersection of sets $\phi \cap \psi$ to create a new belief meeting both requirements.

2.4 Operations and Notation

This section defines some operations involving incomplete beliefs and lists some general notational short-cuts to be used throughout this chapter.

The symbol $\mathcal{P}(G)$ will be used to denote the set of all subsets of set G, i.e., the power set. The symbol $\mathcal{P}_f(G)$ will be used to denote the set of all finite subsets of set G. The set of all functions mapping set F to set G is denoted by $F \to G$ or by G^F.

For complete belief $p \in P$ the expectation of function f is denoted by $p[f]$. For incomplete belief $\phi \in \Phi$ – remember that $\Phi = \mathcal{P}(P)$ – the expectation of function f is denoted by $\phi[f]$ and is defined iff $\forall p, q \in \phi,\ p[f] = q[f]$.

For incomplete belief $\phi \in \Phi$, round brackets will be used as with any probability function: For variables (or events) X and Y, the value $\phi(Y)$ will be defined iff $\forall p, q \in \phi,\ p(Y) = q(Y)$. Similarly, $\phi(Y|X)$ will be defined iff $\forall p, q \in \phi,\ p(Y|X) = q(Y|X)$.

The symbol $\mathcal{R}$ will be used to denote the set of discrete partitions of the elementary

event set Ω where for each $X \in \mathcal{R}$ each $x \in X$ is also a member of event set E. This is the set of discrete random variables for E. The dot product will be used to denote merging two variables,

$$\forall X, Y \in \mathcal{R},\ X \cdot Y = \{\, e \in E \mid \exists x \in X,\ \exists y \in Y,\ e = x \cap y \neq \emptyset \,\} \in \mathcal{R}\ . \tag{2.1}$$

The symbol "$\vdash$" will be used to denote when events, variables or beliefs *determine* each other. Let $X \in \mathcal{R}$ while $e \in E$, the relation $X \vdash e$ (read X determines e) is true iff knowing the outcome of variable X entails a single value for event e. Similarly, when $X, Y \in \mathcal{R}$, the relation $X \vdash Y$ is true iff knowing the outcome of X entails an outcome for Y. For events $e, d \in E$, $e \vdash d$ iff $e \subseteq d$. Beliefs can also determine events and variables. Let $p \in P$ be a complete belief while $\phi \in \Phi$ is an incomplete belief. $p \vdash X$ is true iff and p entails exactly one outcome for variable X. $\phi \vdash X$ is true iff $\phi(X)$ is defined and allows exactly one outcome for variable X.

The symbol "$\models$" will be used to denote when an incomplete belief *models* a variable or event. Let $\phi \in \Phi$ be some incomplete belief while $X \in \mathcal{R}$ is a random variable. The relation $\phi \models X$ (read ϕ models X) is true iff $\forall x \in X,\ \forall p, q \in \phi,\ p(x) = q(x)$. Let f be a function of the form $\Omega \to \mathbb{R}$, relation $\phi \models f$ is true iff $\phi[f]$ is defined, otherwise we write $\neg(\phi \models f)$.

Next, the operation $\oplus : \Phi \times E \to \Phi$ will be defined. For incomplete belief $\phi \in \Phi$ and event $e \in E$ the term $\phi \oplus e$ (read ϕ given e) is defined as,

$$\phi \oplus e\ =\ \{\, p \in P \mid \exists q \in \phi,\ p(\cdot) = q(\cdot|e) \,\}\ . \tag{2.2}$$

Operation $\phi \oplus e$ replaces each member of ϕ with itself conditioned on event e – unless that member rules out e. This is analogous to conditional probabilities for complete beliefs. Building on this, the operation $\otimes : \mathcal{P}(\Phi) \times \mathcal{P}(E) \to \mathcal{P}(\Phi)$ is defined as,

$$\Gamma \otimes D\ =\ \{\, \phi \in \Phi \mid \exists d \in D,\ \exists \gamma \in \Gamma,\ \phi = \gamma \oplus d \neq \emptyset \,\}\ . \tag{2.3}$$

Here $\Gamma \subseteq \Phi$ is any set of incomplete beliefs while $D \subseteq E$ is any set of events. The set $\Gamma \otimes D$ includes those incomplete beliefs that may be obtained by first assuming some $\gamma \in \Gamma$ and then learning some event $d \in D$. It can be shown that,

$$(\phi \oplus e) \oplus d = \phi \oplus (e \cap d) = (\phi \oplus d) \oplus e \tag{2.4}$$

$$(\Gamma \otimes D) \otimes G = \Gamma \otimes (D \cdot G) = (\Gamma \otimes G) \otimes D \tag{2.5}$$

See appendix A.1 and A.2 for proof outlines of equations 2.4 and 2.5 respectively.

It is possible to divide any incomplete belief into two distinct parts which we will call the *uncertain part* and *certain part*. For incomplete belief $\phi \in \Phi$, the certain part is an event denoted by $\epsilon(\phi) \in E$ and is defined as,

$$\epsilon(\phi) = \bigcap \{\, e \in E \mid \phi(e) = 1 \,\} \,. \tag{2.6}$$

The certain part is the most specific event that is believed with certainty. The uncertain part of ϕ is an incomplete belief denoted by $\varphi(\phi) \in \Phi$ and is defined as,

$$\varphi(\phi) = \{\, p \in P \mid p(\cdot|\epsilon(\phi)) \in \phi \,\} \,. \tag{2.7}$$

It should be easy to see that $\varphi(\phi) \oplus \epsilon(\phi) = \phi$; the uncertain part $\varphi(\phi)$ is the vaguest belief (largest member of Φ) that has this property under the condition that no $p \in \phi$ rules out $\epsilon(\phi)$.

Next, we define an operation that *restricts* a belief to a more limited scope. Operation $\phi \wr X$ (read ϕ restricted to X) is defined as,

$$\phi \wr X = \{\, p \in P \mid \exists q \in \phi,\ q(X) = p(X) \,\} \,. \tag{2.8}$$

The restricts relation removes all beliefs about entities not directly captured by the second argument. Finally for $\phi, \psi \in \Phi$, belief ϕ will be called a *vague* version of ψ iff $\psi \subset \phi$.

2.5 The Acting Agents

For the remaining sections of this chapter, assume that there is some set of elementary events Ω and set of events E defined. These events give the acting agents something common to have beliefs about. The event set E should only contain members which are observable. Unobservable parameters might be used to define prior distributions for data sources, or might exist in the minds of acting agents, but E represents the observable world only. $\mathcal{R}$ will denote the set of all discrete partitions of Ω where, for each variable $Z \in \mathcal{R}$, each $z \in Z$ is also a member of E. The set Ω might be uncountable, only members of $\mathcal{R}$ may be observed. This ensures that agents and data sources cannot observe infinite amounts of information.

The set of all valid acting agents that can be defined for this world will be denoted by $\mathcal{A}$. What a valid acting agent is will be described in this section (2.5) and section 2.10. For each agent $i \in \mathcal{A}$, a single *environment* $\mathcal{E}_i = (B_i, u_i)$ is defined. Here B_i is the set of *behaviours* that are defined for environment $\mathcal{E}_i$ while u_i is a *utility function* of the form $\Omega \times B_i \rightarrow [\check{i}, \hat{i}\,]$ with $\check{i}, \hat{i} \in \mathbb{R}$ being lower and upper utility bounds respectively. The expected utility for agent i, when adopting behaviour $b \in B_i$, when the true world state is $\omega \in \Omega$, is then $u_i(\omega, b)$. The acting constraint of agent i will be denoted by U_i, the communication constraint by C_i, and the inference constraint by T_i. The acting constraint formulation is defined in this section. The communication and inference constraint formulations are defined in section 2.10.

Let P be the *base belief set*, the set of all probability functions over E. Let the expected utility of behaviour $b \in B_i$ in environment $\mathcal{E}_i$ under assumption $p \in P$ be denoted by $r_i(b, p)$ where,

$$r_i(b, p) = \int_{\Omega} p(\omega) u_i(\omega, b)\, d\omega \; . \tag{2.9}$$

It is required that all environments be defined such that for all $p \in P$ and $b \in B_i$ the expected utility $r_i(b, p)$ converges. Environments do not include any useless behaviours; it is required that each environment $\mathcal{E}_i$ is defined such that,

$$\forall b_1 \in B_i,\ \neg\exists b_2 \in B_i,\ \forall \omega \in \Omega,\ b_1 \neq b_2 \wedge u_i(\omega, b_1) \leq u_i(\omega, b_2) \; . \tag{2.10}$$

The set of all incomplete beliefs based on P are used as the language of acting agent beliefs and is denoted by $\Phi = \mathcal{P}(P)$. This allows for a very wide range of beliefs, or the absence of beliefs, to be represented (see section 2.3). Members of base belief set P will be called complete beliefs while members of Φ will be called incomplete beliefs.

For a given environment $\mathcal{E}_i$ and incomplete belief ϕ, the preference partial ordering over behaviours B_i might not be complete enough for an optimal behaviour set to be defined. This can happen when ϕ is too vague for $\mathcal{E}_i$. There might be subsets of B_i for which an optimal behaviour set exists. For $B' \subseteq B_i$ to have this property under ϕ, the following must be true,

$$\exists B'' \subseteq B',\ \forall b_1 \in B'',\ \forall b_2 \in B',\ \forall p \in \phi,\ \ B'' \neq \emptyset\ \wedge\ r_i(b_1, p) \geq r_i(b_2, p)\ . \tag{2.11}$$

Note that when ϕ has only one member, $\phi = \{p\}$ where $p \in P$, every subset of B_i is totally ordered. Even in that case, there might not be a subset of B_i for which an optimal behaviour set exists. This can happen when the optimal utility under some p exists only as a limit; where for any $b_1 \in B'$ there is always some $b_2 \in B'$ which is closer to the optimal. To deal with this possibility the requirement for "optimal" behaviour must be relaxed. Imagine that there is some quantity $\delta \geq 0$ of utility by which a behaviour may be suboptimal but still be considered acceptable. For a given $i \in \mathcal{A}$, $B' \subseteq B_i$, $\phi \in \Phi$ and $\delta \geq 0$, this property is defined as,

$$\exists B'' \subseteq B',\ \forall b_1 \in B'',\ \forall b_2 \in B',\ \forall p \in \phi,\ \ B'' \neq \emptyset\ \wedge\ r_i(b_1, p) + \delta \geq r_i(b_2, p)\ . \tag{2.12}$$

When B' holds this property, it will be called a *δ-usable behaviour set* for $\mathcal{E}_i$ under ϕ. For a given $\phi \in \Phi$ there might still be behaviour sets which are not ordered enough to be δ-usable for any $\delta > 0$, but this will happen because ϕ is vague (contains more than one member) and not because the optimal utility is unachievable for some $p \in \phi$. For a given $B' \subseteq B_i$, the set of all $\phi \in \Phi$ under which B' is δ-usable for $\mathcal{E}_i$ will be denoted by $\kappa_i(B', \delta) \subseteq \Phi$. $\kappa_i(B', \delta)$ will be referred to as the *δ-secure belief set* for B'. The value δ will be called an *acting tolerance*.

The purpose of κ_i might be difficult to grasp at first so an example will now be given to illustrate. Imagine that agent i is a competitor in some mathematics competition and

is given the challenge, "Elect to write down either the digits of constant π or of natural number e. The prize will be 1000\$ multiplied by $(1 - 2^{-n})$ where n is the number of correct digits written down. There is no prize if some digit is incorrect. Assume that i has no practical limitations (infinite time and processing power). Assume also that there is no minimum division of prize money. Let b_1 represent the response "$\pi \approx 3.14$" while b_2 is "$\pi \approx 3.141$", b_3 is "$\pi \approx 3.147$" and b_4 is "$e \approx 2.7182$". Let ϕ_1 represent a belief under which neither the definitions nor values of π or e are known. Let ϕ_2 represent a memorization of the first 10 digits of π. Let ϕ_3 represent a memorization of the first 3 digits of π, i.e. 3.14. Let ϕ_4 represent knowledge sufficient to allow an unlimited agent to calculate any finite number of leading digits of π. Let ϕ_5 represent knowledge sufficient to allow an unlimited agent to calculate any finite number of leading digits of e. Let $B' = \{b_1, b_2, b_3\}$ while $B'' = \{b_1, b_2, b_3, b_4\}$. $\kappa_i(B', 0) \subseteq \Phi$ is the set of all beliefs which say enough about π for preference over members of B' to be defined. ϕ_1 is not a member of $\kappa_i(B', 0)$ because an agent holding ϕ_1 does not know what π is and cannot say which behaviors are best. ϕ_2 is in $\kappa_i(B', 0)$ because it can be used to say that b_2 is better than b_1 and b_1 is better than b_3. ϕ_3 is not in $\kappa_i(B', 0)$ because it does not define preference between any of the members of B'. ϕ_4 is in $\kappa_i(B', 0)$ because, like ϕ_2, it totally orders members of B'. ϕ_4 is not a member of $\kappa_i(B'', 0)$ because it does not say what e is so preference between b_2 and b_4 is not defined. ϕ_5 is a member of $\kappa_i(B'', 0)$ because, even though it does not define preference between b_1, b_2 and b_3, it does say that b_4 is better than all three of those, so an optimal behavior from B'' is defined under ϕ_5. Finally, let B''' represent the set of all finite digit estimations of π. The set $\kappa_i(B''', 0)$ cannot contain ϕ_4 because no matter how many digits some $b_j \in B'''$ gives correctly according to ϕ_4, there is some $b_k \in B'''$ that gives even more. For $\delta > 0$ the set $\kappa_i(B''', \delta)$ will contain ϕ_4 because there is some number of digits past which utility gained by going further is less than δ. $\kappa_i(B''', \delta)$ will then contain all beliefs that defines that many digits of π. As δ approaches 0, the set $\kappa_i(B''', \delta)$ converges on those beliefs that give an exact definition for π. Note that ϕ does not have to be correct about the value of π to be a member of $\kappa_i(B''', \delta)$, it merely needs to claim to be. The belief that $\pi = 0$ is more certain than the belief $\pi = 3.1415 \pm 0.0009$. There will be some $\delta > 0$ small enough that belief $\pi = 3.1415 \pm 0.0009$ is not in $\kappa_i(B''', \delta)$ while belief $\pi = 0$ still is. Finally, note that κ_i is not defined relative to practical limitations, it is concerned with the relation between environments and beliefs as considered by some unconstrained ideal agent.

The class of acting constraints that may apply to acting agents can now be formulated. Remember that acting constraints limit an agent's ability to hold beliefs and to turn them into behaviour. For this model, the acting constraints simply put a hard limit on the behaviours that may be considered for a given incomplete belief, and on the incomplete beliefs that may be held. Let $U_i \in \big(\Phi \to \mathcal{P}(B_i)\big)$ denote the acting constraint on agent i. $\Phi \to \mathcal{P}(B_i)$ is the set of all functions mapping members of Φ to subsets of B_i. For a given $\phi \in \Phi$ and $\delta \geq 0$, $U_i(\phi)$ is the set of behaviours that the agent will choose a δ-optimal behaviour from (see section 1.22 for the reasoning behind this). For U_i to be a valid acting constraint, the following condition must hold,

$$\forall \phi \in \Phi,\ U_i(\phi) = \emptyset \ \vee\ \big(\forall \delta > 0,\ \phi \in \kappa_i(U_i(\phi), \delta)\big)\ . \tag{2.13}$$

If U_i did not hold this property, there would be some $\phi \in \Phi$ under which the behaviour set $U_i(\phi)$ does not have a δ-optimal subset for some $\delta > 0$. The agent would have no way to choose a behaviour. Finally, it is required that for all agents $i \in \mathcal{A}$ there exist some $\phi \in \Phi$ such that $U_i(\phi) \neq \emptyset$ – i.e., only agents which can hold at least one belief and which can adopt at least one behaviour are allowed.

The acting constraint U_i gives a flexible language for restricting how an acting agent may turn belief into action. To disallow the agent from holding belief ϕ, choose an acting constraint for which $U_i(\phi) = \emptyset$. To restrict the behaviours available to the agent when it believes ϕ, restrict $U_i(\phi)$.

The set of holdable beliefs for agent i will be denoted by $\Phi_i \subseteq \Phi$; it will be referred to as the *conclusion language* of agent i. The set of *adoptable behaviours* for agent i will be denoted by Δ_i. These are defined as,

$$\Phi_i = \{\, \phi \in \Phi \mid U_i(\phi) \neq \emptyset \,\}\ , \tag{2.14}$$

$$\Delta_i = \bigcup_{\phi \in \Phi} U_i(\phi)\ . \tag{2.15}$$

Acting agent i cannot choose a behaviour when holding beliefs not in Φ_i. Notice that an acting constraint is defined relative to some environment. The acting constraint U_i for environment $\mathcal{E}_i$ can only apply to another environment $\mathcal{E}_j$ when, for all $\phi \in \Phi$, $U_i(\phi)$ is

also a valid behaviour set for $\mathcal{E}_j$ under ϕ – which implies that the adoptable behaviour set must be defined for both environments, i.e., $B_i \supseteq \Delta_i = \Delta_j \subseteq B_j$.

If incomplete belief ϕ is held by the acting agent then the best behaviour must be chosen from $U_i(\phi)$. For a given $\delta \geq 0$, the set of all such optimal allowed behaviours will be denoted by $V_i(\phi, \delta) \subseteq B_i$ and is defined as,

$$V_i(\phi, \delta) = \{\, b_1 \in U_i(\phi) \mid \forall p \in \phi,\ \forall b_2 \in U_i(\phi),\ r_i(b_1, p) + \delta \geq r_i(b_2, p) \,\}\ . \tag{2.16}$$

Given the requirement from equation 2.13, it follows that $V_i(\phi, \delta) \neq \emptyset$ if $\phi \in \Phi_i$ and $\delta > 0$. One way to think of V_i is that, when acting agent i believes ϕ, it will adopt some behaviour from the set $V_i(\phi, \delta)$ for some $\delta \geq 0$, but we cannot know which and the agent has no preference ($V_i(\phi, 0)$ might have more than one member). If it did have a preference, then i would believe more than just ϕ, or U_i would be misspecified; violating the relation between behaviour, belief meaning and acting constraints laid out in section 1.22.

Acting agents, environments, acting constraints and optimal behaviour under acting constraints have all been defined in this section. Our goal has been to allow for many types of acting agents – ideal or less ideal – to be considered in many types of environments. The acting constraints $U_i \in \big(\, \Phi \to \mathcal{P}(B_i) \,\big)$ allow us to separate restrictions on an agent's mind from its environment formulation. This allows us to consider the effects of changing agents or environments separately, for example, by setting $U_i \neq U_j \wedge \mathcal{E}_i = \mathcal{E}_j$ for agents i and j.

Notice that acting constraints are modelled here as hard (qualitative) constraints. In reality, there is little reason to assume that only hard constraints can apply to acting agents. One could make the constraints soft (quantitative) by assigning some utility to the thinking process of an acting agent when turning belief into behaviour. While that would be a realistic extension to our model, we have found that approach difficult to work with.

2.6 Inductive Inference as Belief Translation

Imagine that there is some inference agent which holds complete belief $\sigma \in P$ and must communicate it to acting agent i. Assume that there is no constraint on the inference agent or how it can communicate, it can choose to communicate any belief from Φ_i to acting agent i. The belief σ will be called the *premise*. The set Φ_i is the conclusion language of agent i and is determined by its acting constraint (see equation 2.14). The belief that the inference agent sends the acting agent will be called the *conclusion*. Here, the act of inference is a translation of premise to conclusion. The inference agent must consider acting constraint U_i and environment $\mathcal{E}_i$ when translating.

The inference agent holds premise $\sigma \in P$ and should communicate a conclusion $\phi \in \Phi_i$ which will lead to optimal behaviour under assumption σ. If the inference agent sends $\phi \in \Phi_i$ as conclusion, the acting agent will adopt some behaviour from $V_i(\phi, \delta)$ for some $\delta \geq 0$ (see equation 2.16). It will be assumed that the inference agent does not know δ but does know the environment $\mathcal{E}_i$ and acting constraint U_i. If the inference agent were to send conclusion ϕ and then communicate which behaviour from $V_i(\phi, \delta)$ is better to adopt, then it would be communicating more than just ϕ. Such additional communication will be disallowed since it would mean that the conclusion and acting constraint were misspecified.

Let $s_i(\sigma, \phi)$ denote the value of inference $(\sigma, \phi) \in P \times \Phi$. Since no weighting is defined over members of $V_i(\phi, \delta)$ – even for $V_i(\phi, 0)$ – we are forced to use worst-case expected utility to define the value of an inference (equation 2.17). The value $s_i(\sigma, \phi)$ is defined iff $\phi \in \Phi_i$.

$$s_i(\sigma, \phi) = \lim_{\delta \to 0} \inf_{b \in V_i(\phi, \delta)} r_i(b, \sigma) \tag{2.17}$$

Remember that $r_i(b, \sigma)$ was defined as the expected utility of behaviour b under assumption σ (equation 2.9). The inference agent can now choose a best conclusion from Φ_i by selecting the one that maximises $s_i(\sigma, \phi)$.

For a given premise σ, a conclusion $\phi \in \Phi_i$ which is strictly optimal for σ, might not exist. A conclusion must therefore be found for which $s_i(\sigma, \phi)$ is within $\eta \geq 0$ units of utility from optimal. For premise σ and agent i, the set of η-optimal conclusions will be

called the *solution set* or *η-optimal conclusion set* and be denoted by $S_i(\sigma, \eta) \subseteq \Phi_i$. It is defined as,

$$S_i(\sigma, \eta) = \{ \phi \in \Phi_i \mid s_i(\sigma, \phi) + \eta \geq \bar{s}_i(\sigma) \} , \tag{2.18}$$

where

$$\bar{s}_i(\sigma) = \sup_{\phi \in \Phi_i} s_i(\sigma, \phi) . \tag{2.19}$$

The value η will be called the *inference tolerance* while the value δ will be called the *acting tolerance*. Solution set $S_i(\sigma, \eta)$ will be non-empty for all $\sigma \in P$ and $\eta > 0$ because $\exists \phi \in \Phi,\ U_i(\phi) \neq \emptyset$ was required for U_i to be valid: the agent must be able to hold at least one belief that can be acted on for any $\delta > 0$. Inference (σ, ϕ) will be called a solution for, or η-optimal for, agent i, when $\phi \in S_i(\sigma, \eta)$.

The reader might wonder why only complete beliefs are allowed as premises. The model could be extended to allow incomplete beliefs as premises. It is doubtful that this would add anything useful to the model, which is why it was kept simple.

2.7 Environment Types

It will help to label those environments for which the utility can be calculated for each behaviour from some discrete observation. For this we define *ending environments* and *unending environments* (definition 12). First, minimal sufficient statistics are defined for environments (definition 11).

Definition 11. *Let Z be a partition of Ω while i is some agent. Z is a* **sufficient statistic** *for $\mathcal{E}_i$ iff, for all $b \in B_i$, Z is a sufficient statistic for $u_i(\cdot, b)$. Z is also a* **minimal sufficient statistic** *for $\mathcal{E}_i$ iff every other sufficient statistic Z' for $\mathcal{E}_i$ determines its value, $Z' \vdash Z$.*

Definition 12. *An environment $\mathcal{E}_i$ is* **ending** *iff its minimal sufficient statistic is a member of $\mathcal{R}$. An environment is* **unending** *iff it is not ending.*

Definition 13. *Environment $\mathcal{E}_i$ is called* **behaviourally discrete** *iff B_i is countable.*

Remember that $Z' \vdash Z$ means that the value of Z' determines Z (section 2.4) while $\mathcal{R}$ denotes the set of all discrete random variables. For a environment's utility function to be implementable by some Turing equivalent machine, it must be ending and behaviourally discrete; this is necessary but not sufficient.

As an example of an unending environment, consider an infinite sequence of zeros and ones – as is used for Bernoulli trials. The elementary event set will be $\Omega = \{0,1\}^{\mathbb{N}}$ where $\mathbb{N}$ is the set of natural numbers. Here, it is not possible to define the event set E – i.e. define the σ-algebra – such that all members of Ω are also events. It is possible to define E such that all finite substrings or finite combinations of finite substrings are events. For $\Omega = \{0,1\}^{\mathbb{N}}$, the set $\mathcal{R}$ of discrete observations cannot contain a member which can observe all information.

Environments can be defined which depend on infinite information; as long as the utility converges to a finite amount for each elementary event. Imagine, for example, a utility function that is weighted by a factor that decreases geometrically for consecutive Bernoulli trials. As an example, the function $\sum_{n\in\mathbb{N}} \omega_n \left(\frac{1}{2}\right)^n$ depends on infinite information, but it is computable to within arbitrary precision.

2.8 Data Sources

The expected worst-case utility has been defined for individual inferences in section 2.6. Now the expected worst-case utility for inference algorithms will be considered. For this expectation to exist, the probability of a given premise being produced must be defined.

Data sources are defined for this model as is usual for Bayesian data models. Let Θ be the set of hypotheses. Θ is an unobservable parameter – not corresponding directly to any events from E – and may be uncountable. A prior probability is defined over Θ and is denoted by h. A discrete observation variable $X \in \mathcal{R}$ is defined. A likelihood function $f(E;\Theta)$ is defined over all events from E parametrised by Θ.

The equivalent parameterless distribution is denoted by $\lambda \in P$ and is defined as,

$$\lambda = \int_{\Theta} h(\theta) f(\cdot\,;\theta)\, d\theta \,. \tag{2.20}$$

For each observation variable outcome $x \in X$ there is a corresponding premise $\sigma = \lambda(\cdot|x) \in P$. The set of all such premises will be denoted by Λ and is defined as,

$$\Lambda = \{\, \sigma \in P \mid \exists x \in X,\ \sigma = \lambda(\cdot|x) \,\} \,. \tag{2.21}$$

Λ will be referred to as the *premise set*. For convenience a probability function α is defined over the set Λ such that for all $\sigma \in \Lambda$, $\alpha(\sigma) = \lambda(x)$ if $\sigma = \lambda(\cdot|x)$ otherwise $\alpha(\sigma) = 0$.

The collection (Θ, h, f, X) will be referred to as the *parametrised data model*. The collection (λ, X) will be referred to as the *parameterless data model* or simply as the *data source*. The collection (Λ, α) will be referred to as the *premise source*. Note that the outcomes of X are data while the members of Λ are premises as they include data and a general assumption λ. We will also refer to λ as the *data source hypothesis*.

It might be suggested that data sources should be indexed by agent – as was done for environments and practical constraints. While that would be a more accurate representation of agents in reality, there was no need for it at any point in this theses since the questions considered are all focused on how agents with the same data source may need different inference methods due to interest.

2.9 Vague Conclusions

For a given premise σ, acting agent i and inference tolerance η, there will be a set of optimal conclusions $S_i(\sigma, \eta) \subseteq \Phi_i$, as defined by equation 2.18. Let Z be the minimal sufficient statistic for the intended environment $\mathcal{E}_i$ (definition 12). In general, it is not necessarily true that $\phi \in S_i(\sigma, \eta) \Rightarrow \phi \models Z$ – even when $S_i(\sigma, 0) \neq \emptyset$. Remember that $\phi \models Z$ (read ϕ models Z) was defined in section 2.4 to mean that $\forall p, q \in \phi,\ p(Z) = q(Z)$. Members of $S_i(\sigma, \eta)$ do not have to say something that specific. Consider the belief $\psi = \{\, p \in P \mid p(e) > 0.5 \,\}$ where $e \in E$. This is simply the belief that e being true is more probable than e being false. Here $\phi \models e$ is not true; but it is easy to imagine an environment for which this belief is sufficient for decision making. For example, when choosing a team for a standard "right"/"wrong" football-tipping competition.

Another example where vague conclusions are acceptable is with conditional environments. As an example imagine that the acting agent is a program that will be given the age, gender and income of individuals and asked to guess for each if they own a motorcycle. Here, minimal sufficient statistic Z can be divided into two parts $Z = Z_1 \cdot Z_2$. Z_2 is motorcycle ownership while Z_1 is the age/gender/income information. Z_2 must be predicted given Z_1 but Z_1 is not observed by the data source, it will only be known after a conclusion is selected (the individuals in a training set do not appear in the intended environment). Here, only the conditional (Z_2 given Z_1) relation is of interest to the acting agent. Imagine that some complete belief $q \in P$ has the desired conditional relation. Belief q would say more than is needed. q would define a probability over Z_1 by itself $q(Z_1)$. As a belief, q says more than is needed to act optimally in this environment. For this agent, having a belief about what the distribution of different age/income/gender groups is, is not needed since it will be given Z_1 in its environment anyway. For any $q \in P$ there are infinitely many probability functions in P that share the same conditional relation and therefore entail exact the same behaviour.

If $q \in P$ has the optimal conditional relation then $\phi = \{\, p \in P \mid p(Z_1|Z_2) = q(Z_1|Z_2) \,\}$ is a more vague optimal solution capturing only the desired conditional relation. For problems where the conditional relation is what is of interest, the conditional likelihood is often used as the goodness-of-fit measure (as opposed to the full likelihood function). This is more generally known as conditional learning.

Within a given optimal conclusion set $S_i(\sigma, \eta)$, there may exist conclusions which are maximally vague in this sense: if $\psi \in S_i(\sigma, \eta)$ and there is no $\phi \in S_i(\sigma, \eta)$ such that $\psi \subset \phi$ then ψ is a *maximally vague solution* for σ and η. In general, there could be more than one distinct maximally vague member of $S_i(\sigma, \eta)$ – even for $\eta = 0$.

2.10 Inference and Communication Constraints

Acting constraints have been defined and now inference constraints and communication constraints are considered. For acting agent i, the inference agent must select some inference function from the set ${\Phi_i}^{\Lambda}$. The notation ${\Phi_i}^{\Lambda}$ here represents the set of all functions mapping Λ (the premise set, see section 2.8) to Φ_i (the conclusion language, see section 2.5).

The symbol $T_i \subseteq {\Phi_i}^{\Lambda}$ will be used to denote the inference constraint that applies to inferences for acting agent i. Inference functions that are not members of T_i may not be considered as inference agent solutions.

The inference algorithm design process may be thought of as selecting an algorithm from set T_i. Consider a parameter estimation problem as an example. The inference constraint T_i might include various candidate parameter estimation algorithm implementations, such as: gradient search algorithms, evolutionary algorithms, estimation maximisation (EM) algorithms, or Monte-Carlo Markov chain (MCMC) algorithms. T_i is the set of algorithms that are available – or which one pretends are available when more theoretical goals are adopted.

If the inference agent is to be implemented as a Turing equivalent machine and Λ or Φ_i are infinite, an inference constraints must apply because ${\Phi_i}^{\Lambda}$ would be uncountable. If one wishes to specify, for idealised discussions, that no inference constraint applies, then $T_i = {\Phi_i}^{\Lambda}$ can be written. With the form of inference constraints defined, communication constraints can be considered.

The communication constraint that applies between the inference agent and acting agent i will be denoted by C_i. Before presenting the form that communication constraints take in our model, some examples of communication constraints will be presented in order to illustrate why this choice was made.

One obvious class of constraint on communication which could be imposed is to make some conclusions non-communicable. This could be achieved by further limiting the allowed inference functions to those that do not include the disallowed conclusions in their images (the subset of Φ_i for which there is some member of Λ mapping to it).

Another class of communication constraint could be to limit the number of communicable conclusions. The number of communicable conclusions can be limited to integer n by disallowing all inference functions t for which $|\operatorname{Im}(t)| > n$ (where "Im" denotes the image of a function). One way to imagine this constraint is that before the premise is known, by either agent, a language with no more than n members is constructed specifically for communicating to acting agent i.

Instead of limiting the number of communicable conclusions, the expected amount of information that will be communicated could be limited. For inference function t to

fit an expected information communicated constraint of n bits, the following must hold,

$$\sum_{\phi \in \mathrm{Im}(t)} g(\phi) \log_2 \frac{1}{g(\phi)} \;\leq\; n \;, \tag{2.22}$$

where,

$$g(\phi) = \sum_{\sigma \in \Lambda,\, t(\sigma)=\phi} \alpha(\sigma) \;.$$

Notice that the three classes of communication constraints suggested so far can all be achieved by further limiting the set of allowed inference functions. Perhaps there are possibilities that we have not thought of which cannot be achieved this way, but we will now restrict our model to these types of communication constraints. The communication constraint for agent i will be denoted by $C_i \subseteq {\Phi_i}^{\Lambda}$. If no communication constraint is to apply then $C_i = {\Phi_i}^{\Lambda}$ can be written.

Inference constraints and communication constraints can now easily be combined by intersection of sets. The intersection of these two will be referred to as the inference-communication constraint and abbreviated as *IC constraint*. The symbol K_i will denote the IC constraint for agent i where $K_i = T_i \cap C_i$.

The range of interests that can be expressed by this model has now been defined. A complete interest for acting agent i is composed of: an environment $\mathcal{E}_i$, an acting constraint U_i, an inference constraint T_i, and a communication constraint C_i. Note that a premise source must be specified for T_i and C_i to be definable but not for $\mathcal{E}_i$ and U_i. One might associate $\mathcal{E}_i$ and U_i with the acting agent and associate T_i and C_i with the inference agent.

IC constraints do not apply to individual inferences. The set of optimal conclusions for premise $\sigma \in P$ is still $S_i(\sigma, \eta)$ for agent i and tolerance η (see equation 2.18). When considering a single inference by itself, we need not take into account IC constraints. IC constraints limit what the best inference function is for a premise source (Λ, α) (see section 2.8).

Assume that some interest and premise source have been defined for agent i and that the optimal inference algorithm from the set K_i is sought. If $K_i = {\Phi_i}^{\Lambda}$, the best inference function $k \in K$ can be any function that produces any of the optimal inferences $k(\sigma) \in V_i(\sigma, \delta)$ for every premise $\sigma \in \Lambda$.

When the IC constraint does not allow this ideal inference function, the premise source probabilities α must be considered. For agent i, the best inference function will be defined using expected worst-case utility. Let $w_i(k)$ denote the expected worst-case utility for inference algorithm k; it takes the value,

$$w_i(k) = \sum_{\sigma \in \Lambda} \alpha(\sigma) s_i(\sigma, k(\sigma)) \ . \tag{2.23}$$

Remember that $s_i(\sigma, \phi)$ was defined as the expected worst-case utility for inference (σ, ϕ) (see equation 2.17). The symbol $w_i()$ – here without the parameter – will be used to denote the best achievable worst-case expected utility,

$$w_i() = \sup_{k \in K_i} w_i(k) \ . \tag{2.24}$$

Finally, let $W_i(\iota) \subseteq K_i$ denote the set of ι-optimal inference algorithms for agent i under all its constraints, where $\iota \geq 0$ and,

$$W_i(\iota) = \{ \, k \in K_i \mid w_i(k) + \iota \geq w_i() \, \} \ . \tag{2.25}$$

$W_i(\iota) \neq \emptyset$ is guaranteed when $\iota > 0$ and $K_i \neq \emptyset$.

2.11 Belief Distortion

The model has now been defined and its properties will now be considered. This section looks at the most basic properties of belief distortion (section 1.12, definition 4). First, some acting constraint classes are defined. Remember here that $\kappa_i(B', \delta)$ was defined in section 2.5 as the set of all $\phi \in \Phi$ under which $B' \subseteq B_i$ has a δ-optimal subset for $\mathcal{E}_i$. This property depends on how vague ϕ is – how complete its ordering of B' for $\mathcal{E}_i$ is.

Definition 14. *Acting constraint U_i is* **strict-separable** *iff $\forall \phi \in \Phi_i, \ U_i(\phi) = \Delta_i$.*

Definition 15. *Acting constraint U_i is* **separable** *iff, for all $\phi \in \Phi_i$ there exists no $B' \subseteq \Delta_i$ such that $U_i(\phi) \subset B'$ while $\forall \delta > 0, \ \phi \in \kappa_i(B', \delta)$.*

A strict-separable acting constraint (definition 14) can be "separated" into a conclusion constraint Φ_i and a behaviour constraint Δ_i. Any behaviour that can be considered under one member of Φ_i can be considered under them all. A separable acting constraint which is not strict (definition 15) allows conclusions in Φ_i which do not necessarily define optimal behaviour over all Δ_i; but, it is required that for these conclusions $U_i(\phi)$ be as large as possible within Δ_i.

Definition 16. *Let $B' \subseteq B_i$. U_i is the B'-***restricted** *acting constraint iff, U_i is strict separable, $\Delta_i = B'$, and $\forall \phi \in \Phi,\ \big(\phi \in \Phi_i \ \Leftrightarrow\ \forall \delta > 0,\ \phi \in \kappa_i(B', \delta)\big)$.*

Definition 17. *Let $B' \subseteq B_i$. Acting constraint U_i is B'-***relaxed** *iff,*
$\forall \phi \in \Phi,\ \neg\exists B'' \subseteq B',\ \big(U_i(\phi) \subset B''\ \wedge\ \forall \delta > 0,\ \phi \in \kappa_i(B'', \delta)\big)$.

For a given behaviour restriction $B' \subseteq B_i$, the B'-restricted acting constraint (definition 17) is the largest (having the largest possible Φ_i) strict-separable constraint for which $\Delta_i = B'$. A B'-relaxed acting constraint (definition 17) is a maximal (having the largest possible Φ_i) separable (but not strict) constraint for which $\Delta_i = B'$. There may be more than one for a given B'. The closest we can get to no acting constraints applying is B'_i-relaxed acting constraints. There will tend not to be a single unique B'_i-relaxed acting constraints. For some members of Φ, the ordering over B_i will be too incomplete for a best behaviour to be selected; unless only a subset $U_i(\phi)$ of B_i is considered, compared, and selected from.

Incomplete beliefs which contain only one member always totally order all behaviours. It follows that for all $p \in P$, all restricted and relaxed acting constraints include $\{p\}$ in conclusion language Φ_i.

Belief distortion can now be defined for our model (definitions 18 and 19), and its relation to acting constraints can be described (lemmas 1, 2, 3 and 4, see appendix A.3 for outlines of the proofs).

Definition 18. *An inference (σ, ϕ) is* **belief distorting** *iff $\phi \neq \{\sigma\}$.*

Definition 19. *Acting constraint U_i* **forces belief distortion** *for premise $\sigma \in P$ iff, $\exists \eta > 0,\ \forall \eta' \in (0, \eta],\ \{\sigma\} \notin S_i(\sigma, \eta')$.*

Lemma 1. *It is possible to define an agent $i \in \mathcal{A}$ such that for some $\sigma \in P$, $\{\sigma\} \in \Phi_i$ while U_i forces belief distortion for σ.*

Lemma 2. *If acting constraint U_i forces belief distortion for $\sigma \in P$ then either $\{\sigma\} \notin \Phi_i$ or U_i is not separable.*

Lemma 3. *A restricted acting constraint cannot force belief distortion for any premise, $\forall \sigma \in P,\ \{\sigma\} \in S_i(\sigma, 0)$.*

Lemma 4. *A relaxed acting constraint cannot force belief distortion for any premise, $\forall \sigma \in P,\ \{\sigma\} \in S_i(\sigma, 0)$.*

Lemmas 1, 2, 3 and 4 together say that an acting constraint can force belief distortion but only if the acting agent is incapable of believing the premise, or if the acting agent is incapable of optimising behaviour, within Δ_i, given the premise as conclusion. Remember that conclusion $\{\sigma\}$ is not vague, it totally orders every subset of B_i; there is enough information in $\{\sigma\}$ for behaviour optimisation to be possible to within some tolerance δ.

Just as acting constraints can force belief distortion for a given premise, IC constraints can force belief distortion for a premise source. Definition 20 defines what it means for an inference function to be belief distorting. Definition 21 defines what it means for an interest and premise source to force belief distortion.

Definition 20. *For agent i, inference function $k \in {\Phi_i}^{\Lambda}$ is* **belief distorting** *iff $\exists \sigma \in \Lambda,\ k(\sigma) \neq \{\sigma\}$.*

Definition 21. *The interest of acting agent i is said to* **force belief distortion** *iff there exists some $\iota_1 > 0$ such that for all $\iota_2 \in (0, \iota_1]$ all inference functions $k \in W_i(\iota_2)$ are belief distorting.*

Remember that $W_i(\iota)$ was defined as the set of inference algorithms that are ι-optimal (optimal within ι of the supremum) for premise source (Λ, α) and agent i under all its constraints (equation 2.25). Below, lemma 5 "roughly" says that without practical constraints, belief distortion is never forced, regardless of what the environment is.

Lemma 5. *Let i be some acting agent to which no IC constraint applies ($K_i = {\Phi_i}^{\Lambda}$). Assume that U_i is a relaxed or restricted acting constraint. It follows that the interest does not distort belief.*

Next, how communication constraints may force belief distortion is considered. Definitions 22 and 23 define two types of communication constraints. Lemma 6 describes when these communication constraints may force belief distortion.

Definition 22. *C_i is an n-**discrete communication constraint** iff,*

$$\forall t \in {\Phi_i}^{\Lambda},\ t \in C_i \iff |\operatorname{Im}(t)| \leq n\ .$$

Definition 23. *C_i is an n-**bit communication constraint** iff,*

$$\forall t \in {\Phi_i}^{\Lambda},\ t \in C_i \iff \sum_{\phi \in \operatorname{Im}(t)} g(\phi) \log_2 \frac{1}{g(\phi)} \ \leq\ n\ ,$$

where,

$$g(\phi) = \sum_{\sigma \in \Lambda,\ t(\sigma)=\phi} \alpha(\sigma)\ .$$

Lemma 6. *Assume that no inference constraint applies to agent i ($T_i = {\Phi_i}^{\Lambda}$). Assume that the acting constraint does not force belief distortion for any member of Λ. Let C_i be the n-discrete communication constraint. The interest and premise source can force belief distortion only if $|\Lambda| > n$. Assume, instead, that C_i is the n-bit communication constraint. Now the interest can force belief distortion only if the entropy of α is greater than n bits.*

Lemma 6 says that, by itself, a constraint on the amount of information that can be communicated from inference agent to acting agent, can only force belief distortion if it limits the amount of information to a quantity less than the entropy of the premise source.

2.12 The Order of Environments

For a parameter to be estimated, there are many ways to measure what it is about the parameter that is of value. For example, for the number 7.0009 – for most usual purposes – the last digit (9) would not be valued to the same degree as the first digit (7). It is, nonetheless, possible to imagine a somewhat contrived environment where the last digit is more important.

Loss functions like log-loss or mean-squared error can be used to express what is valued in a parameter. Alternatively, one might choose a parameter-of-interest to estimate and then value all information of the parameter equally. When the parameter is non-finite there might not be a single obvious way to do this. Some methods, like the minimum message length (MML) method, value information contained in a parameter according to how it can help distinguish discrete data points from the set of possible observations. The minimum expected Kullback-Leibler divergence (MEKLD, or minEKL) estimator (as described in [Wallace, 2005, sec. 4.7] and [Dowe et al., 1998]), as well as some Bayesian estimators, value information about the parameter of interest only according to how it can be used to predict the value of some future data variable. These methods all go some way towards avoiding the subjectivity that would come with relying on environments and utilities to define what is valued.

We have chosen to use environments to describe what is of value for a given inference. This goes back to our assumption (section 1.23, under primary assumptions) that the value and meaning of a belief is derived only from how it influences the actions of agents in environments.

Remember that the environment for acting agent i is defined as $\mathcal{E}_i = (B_i, u_i)$ where B_i is the set of behaviours that are defined for $\mathcal{E}_i$ while u_i is a utility function of the form $\Omega \times B_i \to [\,\check{i},\, \hat{i}\,]$ with $\check{i}, \hat{i} \in \mathbb{R}$. This provides a flexible language for describing an acting agent's investment in the truth of propositions. While it might be possible to define agent investment in a way that cannot be described by environments, we suspect that it would be hard to image such possibilities in a real world context. Should a belief be valued for anything other that how it might be acted on?

Two environments might depend on the same information but not value that information equally or in the same way. Can one say, for two environments with the same minimum sufficient statistic, that one of them has a greater – or less-biased – investment in the same information? This is the question that this section, and the next (section 2.13), is concerned with: how the investment of environments may be compared.

Consider now what it means for one environment to be more invested than another. To begin, a partial ordering over a set of environments relative to a given belief will be defined. Let $\sigma \in P$ be some complete belief for event set E. For this section practical constraints will (mostly) be ignored; their effect will be considered in more detail in the

next section (section 2.13).

In section 2.6 the expected worst-case utility for acting agent i adopting incomplete belief $\phi \in \Phi_i$, when the true data source matches complete belief $\sigma \in P$, was denoted by $s_i(\sigma, \phi)$ and defined by equation 2.17. The solution set for agent i under premise σ was then defined as the subset of Φ_i for which $s_i(\sigma, \phi)$ is optimal to within η; it is denoted by $S_i(\sigma, \eta) \subseteq \Phi_i$ (see equation 2.18).

These solution sets can be used to define a partial ordering over the set of agents $\mathcal{A}$. For agents $i, j \in \mathcal{A}$, the relation $i \geq_\sigma j$ will be used to mean that, given premise $\sigma \in P$, conclusions which are good for i are also good for j. More formally, for any $i, j \in \mathcal{A}$ and $\sigma \in P$,

$$i \geq_\sigma j \;\Leftrightarrow\; \forall \eta_1 > 0,\, \exists \eta_2 \geq 0,\, \emptyset \neq S_i(\sigma, \eta_2) \subseteq S_j(\sigma, \eta_1)\ . \tag{2.26}$$

The relation $i \geq_\sigma j$ depends only on: premise σ, the two environments $\mathcal{E}_i$ and $\mathcal{E}_j$, and on the two acting constraints U_i and U_j. The data source and IC constraints (inference-communication constraints, see section 2.5) do not affect it.

Assume that agent i has a B_i-relaxed acting constraint (definition 17). In this case, $\{\sigma\} \in S_i(\sigma, \eta)$ for all $\eta \geq 0$ (see lemma 4). If there is some $\phi \in S_i(\sigma, \eta)$ for all $\eta \geq 0$ where $p \in \phi$ and $p \neq \sigma$, it follows that $\mathcal{E}_i$ cannot distinguish between p and σ – the behaviours that are optimal for σ are also optimal for p. This implies that $\forall q \in \phi,\ \{q\} \in S_i(\sigma, \eta)$ for all $\eta \geq 0$. For such an agent, to know which conclusions are acceptable for a given premise, it is sufficient to know which single member conclusions are acceptable for it. With this, an order relation for environments relative to a given premise can be formulated.

For agents $i, j \in \mathcal{A}$, the relation $\mathcal{E}_i \geq_\sigma \mathcal{E}_j$ will be used to mean that, given premise $\sigma \in P$, conclusions that are good for $\mathcal{E}_i$ are also good for $\mathcal{E}_j$. This relation is not intended to depend on the acting constraints of agents i and j. We will assume that $\Phi_i = \Phi_j = \{\, \{p\} \mid p \in P \,\}$ and that $\forall p \in P,\ U_i(\{p\}) = B_i \wedge U_j(\{p\}) = B_j$. If this were not the case, i and j could be replaced with agents for which it is, while the respective environments stay the same. Relation $\mathcal{E}_i \geq_\sigma \mathcal{E}_j$ is defined as,

$$\mathcal{E}_i \geq_\sigma \mathcal{E}_j \;\Leftrightarrow\; x \geq_\sigma y\ , \tag{2.27}$$

where x, y are any agents for which,

$$\begin{aligned} \mathcal{E}_i = \mathcal{E}_x \ \wedge \ \mathcal{E}_j = \mathcal{E}_y \ \wedge \\ \Phi_x = \Phi_y = \{ \{p\} \mid p \in P \} \ \wedge \\ \forall p \in P, \ \big(U_x(\{p\}) = B_i \wedge U_y(\{p\}) = B_j \big) \ . \end{aligned} \tag{2.28}$$

When $\mathcal{E}_i \geq_\sigma \mathcal{E}_j$ is true, $\mathcal{E}_i$ must be at least as invested in the information contained in σ as $\mathcal{E}_j$. The existence of fully invested environments can now be considered. The symbol $\mathbb{E}^Z$ will be used to denote the set of environments for which random variable Z is the minimal sufficient statistic (definition 11).

Definition 24. *Let Z be some random variable. Environment $\mathcal{E}_i \in \mathbb{E}^Z$ is* fully invested *in Z for premise set $\psi \subseteq P$ iff,*

$$\forall \mathcal{E}_j \in \mathbb{E}^Z, \ \forall \sigma \in \psi, \ \mathcal{E}_i \geq_\sigma \mathcal{E}_j \ .$$

Below, lemma 7 describes one condition under which such a fully invested environment exists. The symbol $\mathbb{E}_d$ will denote the set of behaviourally discrete environments (see section 2.7).

Lemma 7. *Let $Z \in \mathcal{R}$ be some discrete random variable. Let $\psi \subseteq P$ be a set of premises for which there exists some real value $g > 0$ such that, $\forall z \in Z, \ \forall \sigma \in \psi, \ \sigma(z) > g$. There exists some fully invested environment $\mathcal{E}_i \in \mathbb{E}^Z$ such that, $\forall \mathcal{E}_j \in \mathbb{E}^Z, \ \forall \sigma \in \psi, \ \mathcal{E}_i \geq_\sigma \mathcal{E}_j$. This environment will not be behaviourally discrete, i.e., $\mathcal{E}_i \notin \mathbb{E}_d$. See appendix A.4 for a description of how to construct this $\mathcal{E}_i$.*

It should not be too surprising that an environment needs an uncountable amount of behaviours to discriminate between an uncountable amount of beliefs (lemma 7). The requirements that $\forall z \in Z, \ \forall \sigma \in \psi, \ \sigma(z) > g$, is needed because of the requirement from section 2.5 – that expected utility $r_i(b, p)$ converges for all $b \in B_i$ and $p \in P$ – it bounds the achievable utility above and below. Lemma 8 describes the case for behaviourally discrete environments.

Lemma 8. *Let $\mathcal{E}_i, \mathcal{E}_j \in \mathbb{E}_d \cap \mathbb{E}^Z$. There exists some more invested environment $\mathcal{E}_k \in \mathbb{E}_d \cap \mathbb{E}^Z$ such that, $\forall \sigma \in P, \ (\mathcal{E}_k \geq_\sigma \mathcal{E}_i) \wedge (\mathcal{E}_k \geq_\sigma \mathcal{E}_j)$. See appendix A.4 for a description of how to construct this $\mathcal{E}_k$.*

While a fully invested environment for all P does not exist, it is possible to get arbitrarily close – if the environment minimal sufficient statistic is discrete. This suggests that the choice of which environment to target for inductive inference is trivial; any (sufficiently close enough to) fully invested environment with the desired minimal sufficient statistic will do. If some specific environment is of interest, no adaptation would be lost when replacing it with an environment with the same minimal sufficient statistic that is more invested. This would seem to imply that purely statistic-based descriptions of agent investment might be formulated, free of utilities. In the next section (2.13) it will be seen that as soon as practical constraints are considered, this is no longer possible.

2.13 Order Under Constraints

In the previous section (2.12), a partial ordering relation was defined over the set of all valid environments. That ordering did not depend on practical constraints. This section considers the order of environments when practical constraints do apply.

Acting constraints will be considered first. Environments $\mathcal{E}_i$ and $\mathcal{E}_j$ can only be compared under the some acting constraint U_k when U_k can apply to both. The symbol $\mathbb{E}(U_k)$ will be used to denote the set of all environments to which U_k can apply (equation 2.29).

$$\mathbb{E}(U_k) = \{\, \mathcal{E}_i \mid \exists i \in \mathcal{A}, U_i = U_k \,\} \tag{2.29}$$

For acting constraint U_k and premise σ, the partial order relation $\geq_\sigma^{U_k}$ will be defined over the set $\mathbb{E}(U_k)$ (equation 2.30).

$$\forall \mathcal{E}_i, \mathcal{E}_j \in \mathbb{E}(U_k),\ \mathcal{E}_i \geq_\sigma^{U_k} \mathcal{E}_j \ \Leftrightarrow\ x \geq_\sigma y\ , \tag{2.30}$$

where x and y are the agents for which,

$$\mathcal{E}_i = \mathcal{E}_x\ \wedge\ \mathcal{E}_j = \mathcal{E}_y\ \wedge\ U_x = U_y = U_k\ . \tag{2.31}$$

An agent that can be implemented must have a countable set of possible beliefs. Lemma 9 implies that, under any separable acting constraint U_k, with a countable con-

clusion language Φ_k, there will be no fully invested environments; unless, the members of Λ (from the premise source) correspond exactly to the members of Φ_k over the environment minimal sufficient statistic. For acting constraint U_k, the subset of $\mathbb{E}(U_k)$ for which U_k is separable will be denoted by $\mathbb{E}_{\text{sep}}(U_k)$.

Lemma 9. *Let $Z \in \mathcal{R}$ be some discrete random variable. There exists some acting constraint U_k for which Φ_k is countably infinite while there exists some $\sigma \in P$ such that there will exist no $\mathcal{E}_i \in \mathbb{E}^Z \cap \mathbb{E}_{\text{sep}}(U_k)$ such that $\forall \mathcal{E}_j \in \mathbb{E}^Z \cap \mathbb{E}_{\text{sep}}(U_k)$, $\mathcal{E}_i \geq_\sigma^{U_k} \mathcal{E}_j$. If U_k and σ have this property, it will be the case that there exists no $\phi \in \Phi$ for which $\phi(Z) = \sigma(Z)$.*

From lemma 9, it can be see that a fully invested environment does not always exist when, for some $\sigma \in \Lambda$, the acting constraint rules out all conclusions ϕ for which $\sigma(Z) = \phi(Z)$, where Z is the environment minimal sufficient statistic. While fully invested environments will generally not exist under acting constraints, one can look for maximally invested environments. These are environments $\mathcal{E}_i$ for which there exists no $\mathcal{E}_j$, with the same minimal sufficient statistic, such that $\mathcal{E}_i <_\sigma^{U_k} \mathcal{E}_j$. For a given random variable Z, there may be many such maximally invested environments which all differ in how they value the information from Z.

2.14 Acting Agent Hypotheses

Hypotheses sets were defined for data sources in section 2.8. These hypotheses correspond to unobservable aspects of the world. What it means for an acting agent to hold beliefs about unobservable things is considered in this section.

Since event set E was required to only contain observable information, it follows that beliefs from P and Φ do not by themselves say anything about any unobservables. For a given agent i the conclusion language Φ_i is a subset of Φ, so the conclusion language does not explicitly say what the acting agent hypotheses might be.

Consider what would be needed for a candidate set of beliefs to be a valid hypothesis language for a given conclusion language Φ_i. Let Υ_i denote the conclusion hypothesis set for agent i; it is analogous to Θ for the data source. Each member of Υ_i represents a belief about the unobserved entities of the world. What these unobserved entities are is

specific to each acting agent – depending on the form and implementation of its beliefs. It is necessary that each conclusion $\phi \in \Phi_i$ that agent i may hold can be derived from some hypothesis $\gamma \in \Upsilon_i$ by combining it with some observable proposition. This can be done if $\Upsilon_i \subseteq \Phi$; which leads to the requirement,

$$\forall \phi \in \Phi_i,\ \exists \gamma \in \Upsilon_i,\ \exists e \in E,\ \phi = \gamma \oplus e\ . \tag{2.32}$$

Remember (see section 2.4 equation 2.2) that operation $\gamma \oplus e$ replaces each member of incomplete belief γ with itself conditioned on event e – unless that member rules out e. This is analogous to conditional probabilities for complete beliefs.

Clearly, for a given conclusion language Φ_i, there can be many parametrisations Υ_i which meet this requirement. If the agent is to be implemented as a Turing equivalent machine which explicitly keeps track of the unobservable parameter state (as well as an observation value) in order to store a belief, then the set Υ_i would have to be countable.

Remember (see section 2.4 equation 2.3) that the operation $\otimes : \mathcal{P}(\Phi) \times \mathcal{P}(E) \rightarrow \mathcal{P}(\Phi)$ is defined such that $\Gamma \otimes D$ is a set containing those incomplete beliefs that may be obtained by first assuming some $\gamma \in \Gamma$ and then learning some event $d \in D$. $\Gamma \subseteq \Phi$ can be thought of as a set of unobservable propositions while $D \subseteq E$ is a set of observable propositions; $\Gamma \otimes D$ is the language that they form together.

If $(\{\lambda\} \otimes X) \subseteq \Phi_i$ where (λ, X) is the data source (see section 2.8 and equation 2.20), then every $\sigma \in \Lambda$ will also be a member of Φ_i (where (Λ, α) is the corresponding premise source). For Φ_i to make belief distortion unnecessary – for data source (λ, X) and all environments, and assuming no IC constraint applies – it is necessary that $\{\lambda\} \in \Upsilon_i$, and sufficient if the acting constraint is separable. This, in turn, means that for a given data source, Υ_i only needs to contain one member to avoid belief distortion. For a given data source, an ideal acting agent does not need to keep track of some unobservable parameter.

Imagine now that Φ_i is known while the parametrisation Υ_i is not. For a given $\phi \in \Phi_i$ there is something that can be said about the hypothesis $\gamma \in \Upsilon_i$ that generated it,

$$\forall \phi \in \Phi_i,\ \exists \gamma \in \Upsilon_i,\ \exists e \in E,\ \phi = \gamma \oplus e\ \wedge\ \varphi(\gamma) \subseteq \varphi(\phi)\ . \tag{2.33}$$

Recall that $\varphi(\phi)$ is the uncertain part of the belief ϕ (see equation 2.7 section 2.4). It follows from equation 2.33 that, no matter what Υ_i is, any two conclusions $\phi, \psi \in \Phi_i$ could only have been derived from the same member of Υ_i if, $\varphi(\phi) \cap \varphi(\psi) \neq \emptyset$. In other words, if $\varphi(\phi) \cap \varphi(\psi) = \emptyset$ then we know that conclusions ϕ and ψ disagree about some unobservable aspect of the world and cannot be derived from the same hypothesis. For conclusion language Φ_i, every possible agent implementation must keep track of some unobservable parameter iff,

$$\bigcap_{\phi \in \Phi_i} \varphi(\phi) = \emptyset \, . \tag{2.34}$$

The uncertain part $\varphi(\phi)$ of conclusion ϕ is the most vague hypothesis that ϕ could have been derived from. This will be useful in section 2.16 which considers when an inference could be said to forget or create information – including, possibly, unobservable information.

2.15 Alternative Parametrisations

Imagine that the parametrised data model (section 2.8) is (Θ, h, f, X) leading to data source (λ, X). Here Θ is a set of data model hypotheses while λ is a single effectively equivalent data source hypothesis. Similarly, the acting agent has some hypothesis set Υ (section 2.14).

For a given $\theta \in \Theta$ to have a corresponding $\gamma \in \Upsilon$ it must be the case that $\gamma = \{f(\cdot|\theta)\}$. The acting agent can thus have the same hypothesis set as the data model.

It is easy to see that the data model parametrisation Θ is not guaranteed to make the most useful acting agent parametrisation Υ. Assume that it is, now imagine that there is some other data model parametrisation Θ' which leads to the same data source hypothesis λ. Since the same must hold for Θ', it follows that $\Theta' = \Theta$. There can be multiple data models (Θ, h, f, X) leading to the same data source (λ, X); so here is a contradiction.

The data model hypothesis set Θ does not correspond directly to the most useful acting agent hypothesis set Υ. This is simply because, even if some member of Θ is

true, it cannot be observed directly, there will be uncertainty about it. Hypotheses which imitate the distribution over observable world E that is obtained from integrating likelihood f over Θ given X will be more useful.

If the acting agent parametrisation Υ could be chosen free of constraints then the parametrisation that is ideal will correspond to λ (the data source hypothesis). One could simply let $\Upsilon = \{\{\lambda\}\}$.

Imagine instead that $\{\lambda\}$ is ruled out as an acting agent hypothesis due to acting constraints. Imagine that there are two rival acting agent hypothesis sets Υ_i and Υ_j that could be used – representing two possible agent implementations. It would be desirable to know when one is better than the other for a given data source and environment.

Let i and j be two agents sharing a data source and for which $\mathcal{E}_i = \mathcal{E}_j$. Assume that no IC constraint applies to each. Let $D \subseteq E$ where $\Phi_i = \Upsilon_i \otimes D$ while $\Phi_j = \Upsilon_j \otimes D$. Assume that $\forall \phi \in \Phi_i \cap \Phi_j,\ U_i(\phi) = U_j(\phi)$. In this context, we will write $\Upsilon_i \geq \Upsilon_j$ to say that the best solution for agent i is at least as good as the best solution for agent j for all premises $\sigma \in \Lambda$.

Clearly, if $\Upsilon_j \subseteq \Upsilon_i$ then $\Upsilon_i \geq \Upsilon_j$. Simply adding more hypothesis to the conclusion language will not make it worse. It would be more valuable if $\Upsilon_j > \Upsilon_i$, but for this, hypothesis need to be added which are better at imitating λ conditioned on X.

This goal is often achieved using finite weighted summations. These are used, for example, by algorithms using randomised search algorithms; such as evolutionary algorithms and Monte-Carlo Markov Chain (MCMC) algorithms. Even though a hypothesis obtained by weighted summation of members of Θ might not itself be a member of Θ, it can often be a better acting agent hypothesis because it allows for more flexible representation when the premise leaves uncertainty about unobservables.

Another way to extend a given conclusion language to produce a new one is to merge hypotheses. Let Γ be a partition of Θ. We then define a member $\gamma_\nu \in \Upsilon_i$ for each $\nu \in \Gamma$ using the condition,

$$\forall e \in E,\ \gamma_\nu(e) = \frac{\int_\nu f(e;\theta)h(\theta)\ d\theta}{\int_\nu h(\theta)\ d\theta}\ . \tag{2.35}$$

With Υ_i defined as above while Υ_j is defined such that its members correspond directly

to members of Θ, the relation $\Upsilon_i \geq \Upsilon_j$ will hold. The hypothesis set Υ_i can be thought of as being derived from Υ_j by removing (integrating over) some unobservable parameter. By repeatedly removing unobservables from the acting agent hypothesis language in this way, a hypothesis language that has only a single member will eventually be arrived at – it will be equivalent to the data source hypothesis λ. If $\{\{\lambda\}\} > \Upsilon_j$ then it follows that the process of merging hypotheses, as described above, should at some point produce some Υ_i such that $\Upsilon_i > \Upsilon_j$.

2.16 Amplicative and Destructive Induction

In section 1.6, the concepts of destructive and amplicative inference were mentioned and we discussed how applying them to belief as probabilities can be difficult. For all-out logic, destructive inferences forget known propositions while amplicative inferences assume propositions that are not entailed by the premise. In this section, we present definitions of these concepts for our model. Keep in mind, there could be many reasonable ways to define these two inference properties.

Section 2.8 described how a parametrised data model (Θ, h, f, X) entails a unique data source (λ, X) and premise source (Λ, α). For each premise $\sigma \in \Lambda$ there is some $x \in X$ such that $\sigma = \lambda(\cdot|x)$. λ will be referred to as the *source hypothesis*. The parametrised hypothesis set Θ will be ignored in this section.

Section 2.14 described how an acting agent i can be implemented using some *conclusion hypothesis set* denoted by Υ_i. Each conclusion $\phi \in \Phi_i$ is then composed of some unobservable acting agent hypotheses $\gamma \in \Upsilon_i$ and some observation $e \in E$, where $\phi = \gamma \oplus e$.

It helps to separate inference (σ, ϕ) into an observable and unobservable part. The observable part of the premise is the event x. The observable part of the conclusion is the event e. The unobservable part of the premise is source hypothesis λ. The unobservable part of the conclusion is acting agent hypothesis γ.

If both source hypothesis λ and conclusion hypothesis set Υ_i are known, it becomes more easy to reason about amplicative and destructive inferences. If $e = x$ then nothing observable was forgotten or assumed by the inference. If $\neg(x \vdash e)$ then something

observable was assumed. If $\neg(e \vdash x)$ then something observable was forgotten. If $\{\lambda\} = \gamma$ then the unobservable part did not change. If $\lambda \in \gamma$ the unobservable part did change but not in a contradicting way; γ is a more vague version of λ. If $\lambda \notin \gamma$ then the unobservable part did change and not just by limiting the scope of λ. In that case, it makes sense to say that the inference selected some hypothesis.

It would be desirable to define amplicative and destructive inference in a way that is defined the same for a given inference (σ, ϕ), regardless of what the data source and conclusion language is. This can be achieved by giving the agent implementations the benefit of the doubt. If a data source and conclusion language can be defined under which (σ, ϕ) can be said to not be amplicative or destructive, (σ, ϕ) will not be called amplicative or destructive. More precisely, six types of inferences will be defined (definitions 25–30), these are named:

observable-amplicative: *something observable is assumed*

observable-destructive: *something observable is forgotten*

observable-consistent: *nothing observable is assumed or forgotten*

unobservable-amplicative: *something unobservable is assumed*

unobservable-destructive: *something unobservable is forgotten*

unobservable-consistent: *nothing unobservable is assumed or forgotten*

As a shorthand, these will be also be written as: O-Amp, O-Dest, O-Con, U-Amp, U-Dest, and U-Con. The diagram at the end of this section (figure 2.1) shows which combinations of these classes of inference are possible. Note that O-Con rules out both O-Amp and O-Dest while U-Con rules out both U-Amp and U-Dest.

The observable cases are presented first as they are easier to describe (definitions 25, 26 and 27). Recall that for $\phi \in \Phi$ the term $\epsilon(\phi)$ represents the certain part of the belief (see equation 2.6 section 2.4); it is the most specific event that ϕ holds true with certainty, i.e., $\phi(\epsilon(\phi)) = 1$.

Definition 25. *Inference* $(\sigma, \phi) \in P \times \Phi$ *is* **observable-consistent** *(or* **O-Con***) iff,*

$$\epsilon(\phi) = \epsilon(\sigma) \ .$$

Definition 26. *Inference* $(\sigma, \phi) \in P \times \Phi$ *is* **observable-amplicative** *(or* **O-Amp***) iff,*

$$\sigma(\epsilon(\phi)) \neq 1 \ .$$

Definition 27. *Inference* $(\sigma, \phi) \in P \times \Phi$ *is* **observable-destructive** *(or* **O-Dest***) iff,*

$$\phi(\epsilon(\{\sigma\})) \neq 1 \ .$$

Note that it is possible for an inference to be both O-Amp and O-Dest. For the unobservable versions, the relation between the source hypothesis λ and inferred conclusion hypothesis γ will be looked at. Keep in mind that the concern here is with what this relation could be given that only the inference (σ, ϕ) is known.

For the U-Con case it is required that $\gamma = \{\lambda\}$ for some possible data source and acting agent. For the U-Amp case it is required that $\lambda \notin \gamma$ for all possible data sources and acting agents. For the U-Dest case it is required that the inference not be U-Con and that $\lambda \subset \gamma$ (not $\lambda = \gamma$) for some possible data source and acting agent.

Before presenting the unobservable case definitions, an intermediate relation must be presented (lemma 10).

Lemma 10. *Let* $\phi \in \Phi$ *and* $\sigma \in P$*. There exists some* $\lambda \in P$ *and* $\gamma \in \Phi$ *such that* $\lambda \in \gamma$ *where* $\phi = \gamma \oplus \epsilon(\phi)$ *and* $\{\sigma\} = \{\lambda\} \oplus \epsilon(\{\sigma\})$ *iff* $\sigma \in \varphi(\phi) \oplus \epsilon(\{\sigma\})$.

More detail for lemma 10 is given in appendix A.4. With this relation, it is possible to define unobservable amplicative and destructive inference (definitions 28, 29 and 30) without needing to know λ, γ or Υ_i – nor anything about the premise source or acting agent.

Definition 28. *Inference* $(\sigma, \phi) \in P \times \Phi$ *is* **unobservable-consistent** *(or* **U-Con***) iff,*

$$\varphi(\{\sigma\}) \cap \varphi(\phi) \neq \emptyset \ .$$

Definition 29. *Inference* $(\sigma, \phi) \in P \times \Phi$ *is* **unobservable-amplicative** *(or* **U-Amp***) iff,*

$$\sigma \ \notin \ \varphi(\phi) \oplus \epsilon(\{\sigma\}) \ .$$

Definition 30. *Inference* $(\sigma, \phi) \in P \times \Phi$ *is* **unobservable-destructive** *(or* **U-Dest***) iff it is not unobservable-consistent and,*

$$\sigma \ \in \ \varphi(\phi) \oplus \epsilon(\{\sigma\}) \ .$$

The term $\varphi(\phi) \oplus \epsilon(\{\sigma\})$ can be read as, "the uncertain part of the conclusion given the certain part of the premise." The condition in definition 28 will be true iff $\exists\gamma \in \Phi$ for which $\gamma \oplus \epsilon(\phi) = \phi$ and $\gamma \oplus \epsilon(\{\sigma\}) = \{\sigma\}$ – i.e., when there exists some possible γ which can be both the data source hypothesis and the acting agent hypothesis. Applying lemma 10 to definition 29 shows that its condition is met iff no $\gamma \in \Phi$ exists which could be the acting agent hypothesis while containing the data source hypothesis λ. Conversely, the condition for definition 30 can be met iff the opposite is true. Note that U-Con, U-Dest and U-Amp are mutually exclusive.

It follows from these definitions that O-Dest, O-Amp, U-Dest and U-Amp inference are all belief distorting (see definition 18). As such, the environment alone cannot motivate them, there must be practical constraints to explain when and why they are needed (see lemma 5). These four types of induction overlap and divide the space of inferences as shown by the Venn-diagram (figure 2.1).

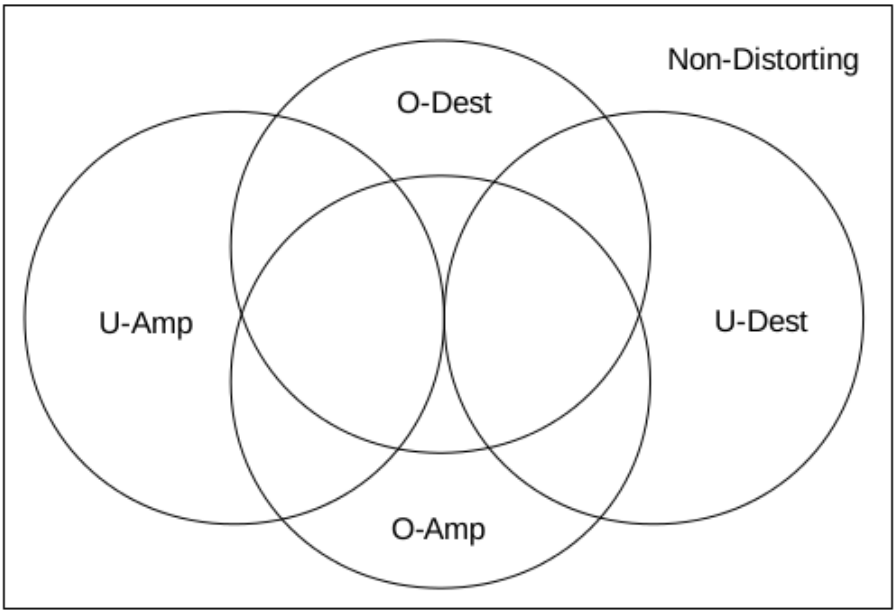


Figure 2.1: types of inference

Note that the set of non-distorting inferences contain exactly those members that are both O-Con and U-Con. They can all be written as $(\sigma, \{\sigma\})$. All other inferences are destructive, or amplicative, or both.

2.17 Constraints Motivating Amplicative and Destructive Inference

In this section, the types of constraints that may force destructive or amplicative inference are considered. Following intuition, it might be suggested that destructiveness could become necessary when a restriction on the amount of communicable information exists. Similarly, it seems reasonable to suggest that a restriction on the amount of communicable information cannot by itself motivate the use of amplicative inference. We have tried to find conditions under which these intuitions hold. Lemmas 11 and 12 give some conditions under which a restriction on the amount of communicable information may force destructive inference but not amplicative inference.

Lemma 11. *Let i be an agent to which an n-bit (or n-discrete) communication constraint applies (see definitions 22 and 23). Assume that the acting constraint is separable (definition 15) and that no inference constraint applies. Let (λ, X) be the data source and (Λ, α) the corresponding premise source. Assume that $\{\{\lambda\}\} \otimes D \subseteq \Phi_i$ where $D = \{\, e \in E \mid \exists d \in \mathcal{P}_f(X),\ e = \bigcup d \ \vee\ e = \Omega \setminus \bigcup d \,\}$. It follows that for all $\iota > 0$ there exists some optimal solution in $W_i(\iota)$ which, for all premises $\sigma \in \Lambda$, might be O-Dest but will not be O-Amp, U-Amp or U-Dest.*

For lemma 12, the condition is relaxed so that only the aspects of the premise that are relevant to the acting agent need be communicated. In this case, U-Dest solutions may be unavoidable.

Lemma 12. *Let i be an agent to which an n-nit (or n-discrete) communication constraint applies (see definitions 22 and 23). Assume that the acting constraint is separable (definition 15) and that no inference constraint applies. Let (λ, X) be the data source and (Λ, α) the corresponding premise source. Let $\gamma \in \Phi$ where $\gamma \in \lambda \wr (Z \cdot X)$ where Z is the minimal sufficient statistic for ending environment $\mathcal{E}_i$. Assume that $\{\gamma\} \otimes D \subseteq \Phi_i$ where $D = \{\, e \in E \mid \exists d \in \mathcal{P}_f(X),\ e = \bigcup d \ \vee\ e = \Omega \setminus \bigcup d \,\}$. It follows that for all $\iota > 0$ there exists some optimal solution in $W_i(\iota)$ which, for all premises $\sigma \in \Lambda$, might be O-Dest and/or U-Dest but will not be O-Amp or U-Amp.*

From lemmas 11 and 12, it can be seen that fairly specific conditions must be met to ensure that restrictions on the amount of communicable information will not force amplicative inference. A proof outline is given in appendix A.6 for these two lemmas. Let's look more carefully at the conditions set by lemmas 11 and 12. The condition $\{\{\lambda\}\} \otimes D \subseteq \Phi_i$ says that it must be possible to select as conclusion any belief which can be composed of the source hypothesis λ and some event $d \in D$ which is a statistic of X. $\mathcal{P}_f(X)$ represents the set of finite subsets of X. The power set $\mathcal{P}(X)$ will be uncountable if X is not finite; $\mathcal{P}_f(X)$ will be countable if X is countable. The condition $\{\{\lambda\}\} \otimes D \subseteq \Phi_i$ cannot be met by any Turing implementable acting agent if D is uncountable.

For the communication constraints considered here, it was assumed that the "code-length" of a given conclusion comes from a conclusion codebook which is optimal for this specific inference problem – this premise source, environment and practical constraint.

Such a situation can only be imagined if it is assumed that both acting agent and inference agent knew, prior to the inference, what experiment X would be performed and what source hypothesis λ would be adopted; and that they both had time to formulate the optimal conclusion codebook.

This optimal codebook situation is not realistic but, it does say what is theoretically possible. Realistically, the codes for communicating conclusions will depend on a common language not designed for the given inference task specifically. For the case where the amount of communicable information is limited and the conclusion codes are fixed in a way not determined by the given inference task, O-Amp and U-Amp inferences are much harder to avoid.

To see the reason, imagine that for each conclusion $\phi \in \Phi_i$ there is a communication cost $l(\phi)$ – measured in information – and that the communication constraint allows only algorithms which map to those members of Φ_i for which $l(\phi) < n$ for some constant n. Conclusions can be can arbitrarily excluded in this way; there seems to be no general way to avoid amplicativity or any other undesirable inference property if the code length function l is not somehow tied to the premise source.

2.18 Description, Prediction and Estimation

In sections 1.24.6 and 1.24.7 related work was discussed where those authors identified and argued for the existence of different types of interests in the practice of experiment design and data analysis. In the language of Shmueli [2010], these include: description, explanation, and prediction. Jebara [2001] uses the terms generative and descriminative similarly to explanation and prediction; he then suggests methods which are hybrid descriminative-generative.

These labels (description, explanation, and prediction) can be thought of as describing properties of an interest. One agent might need an explanation, another needs a predictive algorithm, while a third desires an abbreviated description of the observed data. There may also be agents that require conclusions which are both explanations and predictive methods at the same time. With our model of interest-relative induction, it should be possible to suggest analogous definitions of these interest classes. This section focuses on that goal. There are many ways to define any one of these interest

classes so several definitions are suggest for some here. The purpose of this section is primarily to demonstrate that a formal model of interest-relative induction can be used to describe and investigate the structure of such interest classes.

The concern of this section is in the structure of these interest classes. No attempt will be made here to define these interest properties universally; instead, consideration is given to how they relate to the elements of our model. Several questions must be considered: Can it be known whether a given inference task is predictive (or descriptive or explanatory) just by looking at the environment? Should the data source be considered? Should practical constraints be considered?

Predictive interests will be considered first as they are the easiest to define. An interest is predictive if there is some future data that is relevant to the environment but which might not be observed by the data source.

Definition 31. *Let i be some acting agent. Let (Λ, α) be the premise source. Let Z be the minimal sufficient statistic for $\mathcal{E}_i$. The interest of i is called* **predictive** *iff there exists some $\sigma \in \Lambda$ such that $\neg(\sigma \vdash Z)$.*

Remember that the relation $\sigma \vdash Z$ is true iff complete belief σ leaves no uncertainty about the outcome of random variable Z (see section 2.4). Definition 31 defines interest simply as a relation between the data source and environment; it ignores practical constraints. This seems to be the minimum that is required. Notice that it follows from lemma 5 that the interest being predictive, by itself, cannot motivate belief distortion.

Descriptive interests are considered next. According to Shmueli [2010], the goal of description is to summarise data and represent its structure in a compact manner; it is not aimed at prediction. For now, this interpretation will be adopted to see where it leads.

Assume that agent i has a descriptive interest and that (λ, X) is the data source. It seems reasonable to say that all allowed conclusions record some statistic of X and that they don't make certain statements about events not captured by X – otherwise they would be predicting. The descriptions should also not make an probabilistic statements; no hypothesis or unobservables can be mentioned. The requirement to not make probabilistic statements can be captured as a conclusion language constraint (definition 32).

Definition 32. *Let i be some agent. The conclusion language Φ_i is called* **all-out** *iff* $\forall \phi \in \Phi_i,\ \varphi(\phi) = P$.

See section 2.4 equation 2.7 for the definition of φ. For our model, an *all-out* conclusion language is one that for any event $e \in E$ will either have an exact belief about its truth, or none at all – there are no degrees of belief. The requirement that the inference must record some statistic of the observed data can not be captured by an acting constraint alone because it depends on the relation between observation and inference. This requirement can be captured as an inference constraint (definition 33).

Definition 33. *Let i be some agent. Let (Λ, α) be the premise source. Let $\psi \in \Phi$ be some incomplete belief. The inference constraint $T_i \subseteq {\Phi_i}^{\Lambda}$ is called the* **memorising inference constraint** *under ψ iff,*

$$t \in T_i \ \Leftrightarrow\ \left(\forall \sigma \in \Lambda,\ \exists e \in E,\ t(\sigma) = \psi \oplus e \ \wedge\ \epsilon(\sigma) \vdash e \right).$$

With definition 33, by setting $\psi = P$, it can be ensured that only statistics of the observed data are allowed. This also leads to the reachable conclusion language being all-out (definition 32).

Definitions 32 and 33 can be put together to make a candidate definition for descriptive interests (definition 34). This is not the only candidate definition that will be presented, so we call it *direct descriptive*.

Definition 34. *Let i be some an acting agent. Let (Λ, α) be the premise source. The interest is called* **direct descriptive** *iff $T_i \subseteq T'$ where T' is the memorising inference constraint under P.*

Inferences that are direct descriptive will be either O-Dest or O-Con, will be U-Dest, will not be O-Amp or U-Amp or U-Con (see section 2.16).

An inference must be intended for some use, so consideration must be given to what type of environments can be appropriate for direct descriptive interests. Let Z be the sufficient statistic for environment $\mathcal{E}_i$. Let (λ, X) be the data source. If $X \vdash Z$ then it is at least possible for a purely memorising inference to contain the information of interest. Since direct descriptive interests only allow all-out conclusions, they will say

nothing about unobserved data. If $\neg(X \vdash Z)$ then there is no reason to expect that a simple descriptive interest can produce useful conclusions.

We suggest that the idea that descriptive interests are not conserved with prediction is not strictly correct. A descriptive statistic is used to aid in comprehending data which, in turn, is needed for producing explanations or predictions. The description itself might not say anything about future data but it is intended to aid making judgements about future data. If the descriptive statistic is to abbreviate, it must be known what is of interest, or there would be no way to decide what information is more or less useful.

Imagine that some agent is given a descriptive statistic – one that says nothing about future data – for the agent to make that statistic useful in an environment which is concerned with future data, it must have some prior belief about the world. For this reason, the conclusion that is sent by the inference agent should not be thought of as being the same as the descriptive statistic that was chosen. The inference agent must at least pretend to know what the prior belief of the acting agent is for it to be able to select the best abbreviation. With this taken into consideration, a second definition is suggested for descriptive interests (definition 35).

Definition 35. *Let i be some an acting agent. Let (Λ, α) be the premise source. The interest is called* **indirect descriptive** *iff $T_i \subseteq T'$ for some T' which is the memorising inference constraint under some $\psi \in \Phi$.*

For definition 35, $\psi \in \Phi$ is the prior that the acting agent will combine with the description to act in its environment. The inference agent must at least pretend to know it to be able to abbreviate in a useful way.

As with direct descriptive, inferences that are indirect descriptive will be either O-Dest or O-Con and will not be O-Amp. Here the unobservable property depends on the relation between λ and ψ: If $\{\lambda\} = \psi$ the inference will be U-Con. If $\lambda \subset \psi$ the inference could be U-Dest or U-Con. If $\lambda \notin \psi$ the inference will be U-Amp.

Explanation is the third interest that was mentioned at the start of this section. Explanation is harder to define since it has many interpretations. Explanation will be discussed in more detail in chapter 3 section 3.1. For now, something simpler but related is considered: estimation. Two cases will be dealt with separately: estimation of observable variables, and estimation of unobservable hypotheses. The observable

estimation case is considered first.

An observable estimate must estimate some random variable Z and, if it is to be communicated or to be believed by some Turing equivalent acting agent, then it must be discrete, i.e., $Z \in \mathcal{R}$. Further, an estimate of Z is not really estimation if Z was observed; so the relation between data source and environment must be considered.

Estimation of some unobserved variable Z is belief distorting and cannot be optimal for all environments. Observable estimation is not ideal, it must be justified with a practical constraint. The acting constraint is the most obvious to appeal to. If the agent can only act on beliefs that leave no uncertainty about Z then estimation of Z is necessary. Similarly, communication and inference constraints could also force estimation of Z. Two definitions are suggested to cover both cases (definitions 36 and 37).

Definition 36. *Let i be some acting agent. Let (λ, X) be the data source. The acting constraint U_i is* **observable estimating** *of $Y \in \mathcal{R}$ iff $\neg(X \vdash Y)$ and $\forall \phi \in \Phi_i,\ \phi \vdash Y$.*

Definition 37. *Let i be some acting agent. Let (λ, X) be the data source while (Λ, α) is the corresponding premise source. The IC constraint K_i is* **observable estimating** *of $Y \in \mathcal{R}$ iff $\neg(X \vdash Y)$ and $\forall \sigma \in \Lambda,\ \forall k \in K_i,\ k(\sigma) \vdash Y$.*

Clearly, inferences under these constraints will be O-Amp unless λ is such that $\lambda(Y|x)$ has zero entropy for all $x \in X$.

Next, estimation of unobservable hypotheses is considered. Remember that in section 2.14 it was described how a given conclusion ϕ can be thought of as being composed of an observable part e and an unobservable part γ where $\phi = \gamma \oplus e$.

Ideal inferences can be made when the data hypothesis λ is also allowed as an acting agent hypothesis, i.e., $\{\lambda\} \in \Upsilon$ where Υ is the set of acting agent hypotheses. If this condition is met, the inference agent simply needs to communicate the hypothesis $\{\lambda\}$ and the observation outcome – or some statistic of it if information communication constraints apply (see section 2.16).

For the selection of hypotheses other than the data source hypothesis λ to be forced, there must again be practical constraints. It is known from section 2.16 that a given inference (σ, ϕ) is certain to have selected a hypothesis other than the data hypothesis λ iff it is not U-Con (definition 28). This leaves two possibilities, it was U-Amp or U-Dest.

If it was U-Dest then some more vague version of λ was selected; it would be strange to call that hypothesis selection. If the inference was U-Amp then it selected a hypothesis that is not λ.

With this, it might be suggested that U-Amp inferences are exactly those that select hypothesis. There is a complication here. If there exists some $\gamma \in \Phi$ such that all members of Φ_i can be derived from it, then the inference agent doesn't really select hypothesis γ; γ is assumed by the acting agent before the inference. It therefore makes sense to require that it not be possible to derive the conclusion language from a single hypothesis.

Definition 38. *Let i be some acting agent with data source (λ, X). Conclusion language Φ_i is* **unobservable estimating** *if $\bigcap_{\phi\in\Phi_i} \varphi(\phi) = \emptyset$ and $\forall\phi \in \Phi_i,\ \lambda \notin \varphi(\phi)$.*

The requirement that $\lambda \notin \varphi(\phi)$ states that whatever hypothesis ϕ was derived from, it is not λ or some restricted version of it. The requirement $\bigcap_{\phi\in\Phi_i} \varphi(\phi) = \emptyset$ states that the conclusion language Φ_i cannot be derived from a single hypothesis; no matter how acting agent i is implemented, it must represent more than one unobservable hypothesis (see section 2.14).

Definition 38, unfortunately, does not capture all possible combinations of constraints that may force estimation of unobservable parameters. We have not been able to find a more general definition which is not excessively complex. Definition 39 describes a similar class of IC constraint that will force unobservable hypothesis selection.

Definition 39. *Let i be some acting agent with data source (λ, X) and corresponding premise source (Λ, α). IC constraint K_i is* **unobservable estimating** *if,*
$\bigcap_{\sigma\in\Lambda} \varphi(k(\sigma)) = \emptyset$ and $\forall\sigma \in \Lambda,\ \lambda \notin \varphi(k(\sigma))$.

For both definitions 38 and 39, all inferences will be U-Amp.

Notice that an interest can be predictive, descriptive, observable estimating and unobservable estimating all at the same time or in any combination. Of these four classes, when considered individually: Only observable estimating always forces O-Amp by itself. Only unobservable estimating always forces U-Amp. Only direct descriptive always forces U-Dest. Only indirect descriptive and predictive allow non-distorting solutions.

2.19 Chapter Summary

Section 1.23 listed the assumptions on which our model is built. The purpose of this chapter has been to make clear implications of those assumptions. This section summarises those findings.

Incomplete Beliefs and Acting Constraints:

It is necessary to first briefly describe how acting agent beliefs and acting constraints were formulated. E denotes the *observable event set* based on elementary event set Ω. E is the set of all events that may be directly observed by the data source and within the environments of acting agents. A *complete belief* about the observable world may be specified by defining a probability function p over all E. Complete beliefs define an exact credence for each event $e \in E$. There is assumed to be some *data source* and *premise source*. The data source produces observations from experiment X which is a discrete random variable of E. The premise source has some prior belief λ about the world where λ is a probability function over all E. Premises are produced by combining observations with this prior belief. For observation $x \in X$, the corresponding premise will be $\lambda(\cdot|x)$. The *inference agent* must translate these premises for *acting agents* according to their practical constraints and environments.

Since this work is concerned with limited agents, the acting agents hold beliefs from the set Φ of *incomplete beliefs* for E. The reader should look at section 1.22 to see what incomplete beliefs are and how the acting constraints of acting agents are defined in relation to belief and behaviour. For agent i: its environment is denoted by $\mathcal{E}_i$; the behaviour set defined for $\mathcal{E}_i$ is denoted by B_i; the expected utility of behaviour $b \in B_i$ is $u_i(\omega, b)$ when $\omega \in \Omega$ is the true world state; and the acting constraint is denoted by $U_i : \Phi \to B_i$.

The acting constraint limits the beliefs that an agent may hold; it also limits the behaviours that the agent may consider according to what belief it holds. When the agent believes $\phi \in \Phi$, it must choose a behaviour from $U_i(\phi) \subseteq B_i$. If $U_i(\phi) = \emptyset$, agent i may not hold belief ϕ – i.e., ϕ is not in its *conclusion language* (denoted by Φ_i).

2.19.1 Forced Belief Distortion

The first and most basic results come from section 2.11 where belief distortion was considered in general. Remember that an inference is defined as *belief distorting* (section 1.12, definition 4) iff, for some observable proposition, the conclusion and premise do not make identical predictions.

If – for a given acting constraint, environment and premise – the set of optimal conclusions contains only belief distorting members, the interest is said to *force belief distortion* for that premise. See section 2.11 for the exact definition.

If – for a given interest and premise source – the set of optimal inference algorithms contains only members that will distort belief for some premise, the interest is said to force belief distortion for that premise source. See section 2.11 for the exact definition.

When no practical constraints apply, belief distortion is never forced (lemma 5). That should not be too surprising, the model was set up to make practical constraints the only valid motivation for belief distortion.

Under Acting Constraints:

The types of acting constraints that may lead to belief distortion being forced were considered in section 2.11. In the absence of inference and communication constraints, acting constraints may force belief distortion in two ways: First, if the acting constraint explicitly prohibits the acting agent from holding a belief which is observably equivalent to the premise. Second, if the acting constraint is not separable (lemma 2). *Separable* acting constraints were defined in section 2.11. Put simply, an acting constraint is separable if the restriction on which behaviours the agent may consider is not overly dependent on what belief from its conclusion language it holds. As an example, let ϕ_1 and ϕ_2 be beliefs that the agent is allowed to hold. If the acting constraint allows for behaviour b to be considered when ϕ_1 is held, and the constraint is separable, then b must also be allowed to be considered when ϕ_2 is held, unless ϕ_2 is too vague for the value of b to defined. See definition 15 section 2.11 for the exact definition of separable acting constraints. Two other classes of acting constraints are also listed in that section which will never force belief distortion (lemmas 3 and 4).

Under Information Communication Constraints:

There are many ways in which the process of communicating conclusions can be restricted. Section 2.11 considered constraints on the amount of information that can be communicated from the inference agent to the acting agent. Here it has to be imagined that each conclusion that is allowed for the acting agent has some code and that each code has a code-length. An information communication constraint may then restrict the code-lengths that may be communicated – either absolutely or in expectation (according to the premise source).

If arbitrary code-lengths are allowed there is no guarantee that belief distortion will not be forced. Imagine, instead, that the code-lengths are designed to be optimal for the given premise source and interest. In this case, the information communication constraint – by itself – will force belief distortion iff the data source entropy exceeds the information limit of the communication constraint.

2.19.2 Investment of Environments

Agents in nature – such as humans, animals and computer programs – are concerned with different environments. Intuitively, it makes sense to say that some environments are more invested than others (in the truth of various positive propositions about the world), but there may also be environments which are incomparable. Below is a simplified example.

If one geologist is concerned with finding copper deposits while another is concerned with finding oil wells, their environments are not strictly comparable. The knowledge that is needed for finding copper deposits does not contain all knowledge that is needed for finding oil wells – or vice versa. Imagine now, a third geologist who is concerned with finding both copper deposits and oil wells. The third geologist is strictly more invested than the first two. If the third geologist is perfectly adapted to its environment, it is automatically perfectly adapted to the environments of the first two.

In section 2.12, fully invested environments were considered. An environment is fully invested if being adapted to it guarantees being adapted to any other environment with the same sufficient statistic (being concerned with the same observables). While such

an environment does not strictly exist for our model, it is possible to get arbitrarily close to fully invested environments. If environments with unbounded lower utility and uncountable behaviour sets were allowed, environments based on log likelihood could be defined which are fully invested.

It may seem that using more invested (closer to fully invested) environments is desirable for guiding inferences. Practical constraints interfere with this intuition. In section 2.13 it was shown that when practical constraints are considered, there is not always a more invested environment.

If agents i and j – to which no practical constraints apply – are concerned with environments $\mathcal{E}_i$ and $\mathcal{E}_j$ respectively, there will exist some environment $\mathcal{E}_k$ such that being adapted to $\mathcal{E}_k$ guarantees being adapted to $\mathcal{E}_i$ and $\mathcal{E}_j$. If, instead, a practical constraint does apply, an environment $\mathcal{E}_k$ holding this property might not exist (see lemma 9 in section 2.13). That will only happen when the acting constraint rules out the premise (or something effectively equivalent over the minimal sufficient statistics of $\mathcal{E}_i$ and $\mathcal{E}_j$) as a valid conclusion.

When no practical constraints apply, it is sufficient to know the minimal sufficient statistic of the environment of interest. When practical constraints do apply, fully invested environments do not always exist. As the constraints become more severe, the investment of environments becomes more incomparable and it becomes better to specify more precisely the environment of interest. It becomes better to specify, not only what information is of interest, but how it is valued; which can be expressed more flexibly through utility functions (as opposed to just specifying sufficient statistics). This effect can be caused by acting constraints, communication constraints and inference constraints.

2.19.3 Conclusion Hypothesis Languages

Section 2.15 considered the effect that the language of allowed conclusions has on inference. More specifically, it looks at *acting agent hypotheses* and how they relate to the data source. The allowed beliefs (the conclusion language) of an acting agent must be derived from some acting agent hypothesis set. The idea is that a given conclusion must be represented in the acting agent implementation by two parts: The first part states

what unobservable acting agent hypothesis was selected. The second part states some observable belief; this may include something that was observed by the data source or something that was predicted. A given conclusion $\phi \in \Phi$ is then composed of some unobservable hypothesis $\gamma \in \Upsilon_i$ and some observable event $e \in E$. Here Υ_i denotes the set of allowed acting agent hypotheses for acting agent i. The members of Υ_i are incomplete beliefs, so $\Upsilon_i \subseteq \Phi$.

A prior probability distribution over all observables can be defined by specifying a hypothesis set Θ, a prior probability h over Θ, and a likelihood function $f(E;\Theta)$ which defines a distribution over all observables E for each hypothesis $\theta \in \Theta$. Together, Θ, h and f define a *parametrised model* of the observable world. An equivalent *parameterless model* can be obtained by integrating over Θ. The symbol $\lambda(E)$ is used to denote this parameterless model of the observable world. For each $e \in E$,

$$\lambda(e) = \int_{\Theta} h(\theta) f(e;\theta)\, d\theta\ . \tag{2.36}$$

For Bayesian and algorithmic methods, data source models (prior beliefs about the observable world E) are defined using some hypothesis set Θ as intermediate. Hypothesis simplicity is defined for members of Θ as weightings $h(\Theta)$.

One should not assume that the hypothesis set Θ which is ideal for defining simplicity, is always also ideal for inference. For a given interest, there will be a conclusion language. This is the language of conclusions that can be inferred to under the practical constraint. It is important when considering interest-relative induction, not to confuse the parametrised data model hypothesis set Θ with hypotheses that the acting agent may believe. The acting agent hypotheses are determined by the acting agent implementation. It is possible to define a practical constraint such that for each $\theta \in \Theta$ there is a corresponding equivalent $\gamma \in \Upsilon_i$. This will automatically be the case when no practical constraint applies. Section 2.15 was concerned with the relation between Υ_i, Θ and λ.

Since there are many different parametrised data models (Θ, h, f) which can lead to the same parameterless data model λ, there is no reason to think that optimal inferences will lead to conclusions that correspond to – or which are consistent with – members of Θ. In other words, the acting agent hypotheses from Υ_i that are optimal to select will not necessarily correspond to members of Θ. This is not the effect of practical constraints,

it holds even in their absence. Let X be the variable that is observed by the data source while Y is the variable about which environment $\mathcal{E}_i$ is concerned (the environment minimal sufficient statistic). Even when X and Y are conditionally independent under $f(\cdot\,;\theta)$ for each $\theta \in \Theta$, the optimal inferences will not always select members from Υ_i which correspond to members of Θ – even when there is no practical constraint forbidding it. Hypotheses which imitate the structure of $\lambda(Y|x)$ will be preferred.

There are simple ideal conclusion languages which will not force belief distortion. For those, the acting agent hypothesis set Υ_i only needs to contain one member; one that is derived from the parameterless data model λ. In this case, a hypothesis does not need to be "selected" since there is only one acting agent hypothesis.

Improving the Conclusion Hypothesis Language:

For a given inference task one must first decide which hypotheses will be inferred to (the set of acting agent hypotheses Υ_i). This is part of the act of specifying a practical constraint.

Given some parametrised data model hypothesis set Θ (and corresponding likelihood functions f), it might be possible to use Θ as the acting agent hypothesis set Υ_i. Alternatively, it might be possible to derive an acting agent hypothesis set Υ_i from Θ which will be as good or better. Section 2.15 describes two ways in which this can always be done – regardless of what the intended environment is. Examples of both methods are given in chapter 4.

The first method extends Θ by including hypotheses in Υ_i which are derived from multiple members of Θ by finite weighted summation. Essentially, the acting agent hypothesis set Υ_i selects multiple members of Θ. This method is often used for inference methods which use randomised search algorithms such as Monte-Carlo Markov chain (MCMC) algorithms and evolutionary algorithms.

The second method for extending Θ that was mentioned was merging hypotheses. Here, members of Υ_i are obtained by partitioning Θ and deriving the members of Υ_i from that partition's members. One way this can be achieved is by "removing" parameters from a parametrised data model. As an example, assume that each $\theta \in \Theta$ takes the form (μ, v) where μ is a mean and v is a variance. Instead of selecting a single pair (μ, v)

as the conclusion, one might define Υ_i such that it selects a single value for μ only and then describes belief about v as a distribution derived from the prior h and observed data x.

By designing the conclusion hypothesis language Υ_i to estimate fewer parameters exactly, and to better imitate the structure of $\lambda(E|x)$ (the parameterless data model conditioned on the observation variable), forced belief distortion can be reduced.

2.19.4 Amplicative and Destructive Inferences

There are many ways belief may be distorted when translating a premise to a conclusion. It is desirable to be able to describe different classes of belief distortion. With such classes it becomes possible to say something about which types of practical constraints may force different types of belief distortion.

Amplicative reasoning distorts belief by assuming something that is not entailed by the premise. *Destructive* reasoning distorts belief by forgetting something that is held by the premise. In section 2.16, very general definitions for these concepts were introduced. More specifically, both were divided into two subclasses: observable and unobservable. This gives four general classes of belief distortion: observable-amplicative, unobservable-amplicative, observable-destructive, and unobservable-destructive. These are abbreviated as: O-Amp, U-Amp, O-Dest and U-Dest respectively. O-Amp inferences assume some observable event $e \in E$ which was not entailed by the premise. O-Dest inferences forget some observable event $e \in E$ which was believed by the premise. U-Amp inferences select some unobservable acting agent hypothesis $\gamma \in \Upsilon_i$ which was not entailed by the premise. U-Dest inferences select some unobservable hypothesis $\gamma \in \Upsilon_i$ which is consistent with the premise but does not fully capture all its unobservable aspects. Any inference which does not belong to at least one of these classes, does not distort belief. The four classes do overlap and the Venn diagram below (figure 2.2) shows how they divide the space of possible inferences.

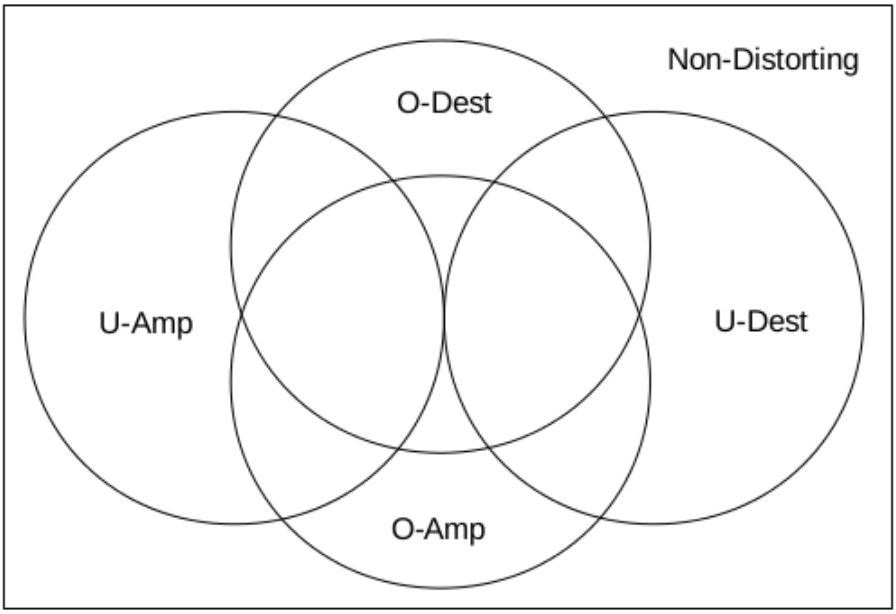


Figure 2.2: types of inference

These classes allow a more intuitive understanding of different types of belief distorting inferences. An O-Dest inference forgets something but the acting agent might recover that information through observation. A U-Dest inference forgets something but the acting agent might recover that belief by making an unobservable assumption. An O-Amp inference assumes some observable information which might be contradicted by future observation. A U-Amp inference makes some unobservable assumption which cannot be absolutely contradicted by observation alone.

Amplicativity and Information Communication Constraints:

In subsection 2.19.1, *information communication constraints* were described. These limit the amount of information that can be communicated from inference agent to acting agent. Intuitively, it makes sense to expect information communication constraints to force the use of inference algorithms which may make destructive inferences but not amplicative inferences. In section 2.17 it was shown that this can be the case; but, the allowed language of conclusions must meet certain (fairly strict) conditions.

Conclusion languages that meet this condition are easy to define but might not be very useful: they may be very difficult to compute predictions of interest from. The

conclusion languages of many common inference methods do not meet this condition. Inference methods which estimate intermediate parameters will not meet this condition; unless they use "conjugate priors" (see [Fink, 1997] for a definition, see appendix B.2 for the relation between conjugate priors and amplicativity).

2.19.5 Three Classes of Interest

In sections 1.24.6 and 1.24.7, related work was discussed where those authors identified and argued for the existence of different types of interests in the practice of experiment design and data analysis. In the language of Shmueli [2010], these include: description, explanation, and prediction. Jebara [2001] uses the terms generative and descriminative similarly to explanation and prediction; he then suggests methods which are hybrid descriminative-generative.

Section 2.18 considered how, within our model, interests can be divided into analogous classes. Three broad classes of interest were considered: *predictive*, *descriptive* and *estimating*. These classes overlap: a given interest may fall into a combination of these.

These classes can be identified by looking at the structure of the interest and its relation to the premise source. By giving exact definitions of these classes, it becomes possible to say what types of interests may force different types of belief distortion. Remember that in section 2.16 different types of belief distortion were defined. These were: O-Amp, U-Amp, O-Dest and U-Dest.

By knowing what interest classes a given interest belongs to (with respect to the premise source), something can be said about how optimal inference algorithms for them will distort belief. The remainder of this subsection summarises those relations.

An interest can be defined as predictive without knowing the practical constraint; only the relation between data source and environment needs to be considered. An interest being predictive, by itself, will not force belief distortion.

Descriptive interests are somewhat hard to define. There are several ways to define this interest class within our model. Descriptive interests will not force O-Amp inference and may force O-Dest inference. Whether it will force U-Dest inference depends on the details of how we choose to define descriptive interest as an interest class. Section 2.18 presents some alternatives. For one of these, U-Amp inference will be forced, but in a

way that is not dependant on the observed data – so the inference cannot be thought of as explaining anything.

Explanation is a difficult concept to define exactly within our model so a related class of interest was looked at: estimation. Two types of estimation were considered: observable estimating, and unobservable estimating. Observable estimating interests force O-Amp inferences. Unobservable estimating interests force U-Amp inferences. Explanation itself will be discussed in section 3.1.

Chapter 3

Implications for Select Topics

3.1 Explanation

3.1.1 Explanation and Prediction

In this subsection it is argued that – when the primary assumptions that went into our model are accepted (section 1.23) – one key difference between explanation and prediction is that explanation must be understood in the context of practical constraints and limited environments of interest.

There are subtle ways in which existing definitions for explanation differ. Explanation is typically characterised as being concerned with saying something about the underlying causes and mechanisms of the phenomenon being studied. We will not propose a complete definition for explanation; instead, a property is specified that will be assumed to always hold for all explaining inference algorithms. If inference function t – which maps premises to conclusions – is *explaining*, it must hold the following property: There is at least one premise σ for which t will be amplicative while the belief so created is not also held by all conclusions that t may produce. Essentially, the part of conclusion $t(\sigma)$ which counts as the "explanation part" cannot be entailed by σ and, t must not select that same explanation regardless of what the premise is. Remember that a premise σ is the combination of prior belief λ and observation outcome x. The reason amplicative inference is required is that any conclusion arrived at by non-amplicative inference can be represented in a predictively equivalent form by simply describing λ

and x. Such conclusions require no explicit statements about unobservable entities or unknown observable entities.

There may be many inference functions which hold this property while not really being explanation methods. The condition is considered necessary but not sufficient. The condition is presented more formally by definition 40.

Definition 40. *Let $t : \Lambda \rightarrow \Phi_i$ be some inference algorithm where Λ is the set of premises that may be produced by the premise source while Φ_i is the language of usable conclusions for acting agent i. If inference function t is* **explaining** *then at least one of the following conditions must be true:*

Case 1: $\neg\exists\gamma \in \Phi,\ \forall\sigma \in \Lambda,\ \exists e \in E,\ \gamma \oplus e = t(\sigma)$.

Case 2: $\neg\exists\sigma_1, \sigma_2 \in \Lambda,\ \exists e \in E,\ \sigma_1(e) < 1\ \wedge\ t(\sigma_1)(e) = 1\ \wedge\ \neg(t(\sigma_2)(e) = 1)$.

Case 1 from definition 40 will hold if t is U-Amp (unobservable-amplicative) for some premise. Case 2 will hold if t is O-Amp (observable-amplicative) for some premise. The classes of U-Amp and O-Amp inferences are defined in section 2.16 (definitions 29 and 26 respectively). To know whether a given inference is U-Amp or O-Amp, it is sufficient and necessary to know what the premise and conclusion is – the interest and premise source does not matter. Both conditions from cases 1 and 2 ensure that the same explanation is not trivially used for all possible premises.

U-Amp and O-Amp inferences are always belief distorting (see section 2.16). From this, two things follow: there must be practical constraints applying to the inference task and, there must be a limited environment of interest. To say why an explanation is desired, as opposed to a non-distorting inference, practical constraints must be appealed to. With practical constraints applying, it follows that there can be no single inference method which conforms to the constraint while being optimal for all environments. This last point was made in more detail in section 2.12 where it was shown that under practical constraints, fully invested environments do not exist.

In section 2.18, a definition (31) for predictive interests was proposed. This property is simply a relation between the premise source and the environment of interest. The environment of interest must depend on some observable proposition which might not be observed by the data source. Prediction as an interest can be defined without mentioning practical constraints. For a data source that conforms exactly to the assumed prior belief,

a single inference algorithm that is optimal for all environments will exist which does not distort belief – it will simply use Bayesian prediction. Practical constraints may still apply for predictive interests – ruling out this ideal inference algorithm – but they are not necessary as with amplicative inferences.

In section 2.18, different classes of interest were defined and related to the different types of belief distortion. Amplicative inference will not be forced because the interest is predictive (definition 31). Thus, explaining inference methods are not motivated by the need for prediction alone. U-Amp inference may be forced by descriptive interests (definitions 34 and 35), but they will then always lead to the same unobservable hypothesis γ being selected. Thus, descriptive interests cannot motivate explanation. Two definitions were proposed for observable estimating interests (definitions 36 and 37), describing a desire for all-out (non-probabilistic) predictions. Neither of these can force U-Amp inference by themselves. Two definitions were proposed for unobservable estimating interests (definitions 38 and 39), describing a desire to estimate unobservable hypothesis. These will force U-Amp inference. They will not in general select the same unobservable hypothesis γ for all observations; though, that depends in the premise source. The desire to estimate unobservables is therefore a valid motivation for explaining inference methods.

In summary: prediction as an interest can be considered without mentioning practical constraints. Practical constraints are essential to motivating explanation. Of the different classes of interests that have been looked at (section 2.18), neither predictive nor descriptive interests require explanation.

3.1.2 Conclusion Brevity Requirements

U-Amp is a specific type of belief distortion, by taking this into consideration more can be said about the types of constraints which may motivate explanation via unobservable hypotheses. This subsection considers constraints requiring explanation brevity. The next subsection (3.1.3) brings together some of our conclusions about valid motivations for explanation.

In section 2.11, constraints on the amount of information that may be communicated between inference agent and acting agent were defined (see definitions 22 and 23). These constraints represent how it is often the requirement that an explanation be brief.

The "length" of a conclusion (measure of how brief it is) will depend on the language that it must be given in. For example, the conclusion of a scientific study might be written in English and using mathematics. The length might then be measured as the number of characters to print. Let's imagine that each possible conclusion ϕ that is considered prior to the inference task being performed has some code length $l(\phi)$. A constraint on the amount of information that can be contained in the conclusion may rule out certain conclusions or inference algorithms and so force belief distortion.

Intuitively, it might make sense to suggest that constraints requiring conclusion brevity can force the use of destructive inference but not amplicative inference. Section 2.17 showed how that intuition only holds under very ideal conditions.

Now, a specific question will be considered: When can or cannot the use of amplicative inference be motivated by conclusion brevity constraints? The relation is complicated by two factors: First, other practical constraints (acting constraints and inference constraints) may complicate the relation. Second, the relation depends on how "optimal" the conclusion codebook (denoted here by l) is for this specific inference task.

In many cases, the codebook l will not be optimised for the specific inference task. For example, the length of a conclusion written in English is determined by the English language which is not optimised specifically for any one inference task. In this case, little can be said about what types of belief distortion might be forced. O-Amp, U-Amp, O-Dest and U-Dest belief distortion can all be forced by constraints on the brevity of conclusions when the codebook is suboptimal. This is true even when no other practical constraints apply (see section 2.17).

If the codebook is optimised for the specific inference task, the brevity constraint will not by itself force amplicative belief distortion (either O-Amp or U-Amp). Even with optimised codebooks, when acting constraints are present, they may interact with brevity constraints and force amplicative belief distortion. That can be avoided if the acting constraints are relaxed enough. One condition that, if met by the acting constraint, will avoid amplicative belief distortion being forced due to such interaction, is given in section 2.17. Under those conditions, only destructiveness (O-Dest and U-Dest) may be forced.

How realistic is it to expect that the measure of conclusion brevity be optimal for a given inference task? This will depend on the extent to which the inference agent and

acting agent may coordinate prior to the inference being made. To answer this question, a somewhat abstract example will be used to illustrate some possibilities.

Imagine that there are several agents all concerned with the relation between disease d and symptom s. It is initially not known if s causes d, is caused by d, is the result of an unobserved third cause, or if there is no causal relation at all. Experiment X will be performed leading to outcome $x \in X$ and an inference agent will produce some conclusion. The inference agent has some prior belief λ about the relation between s and d which will be combined with outcome x to produce conclusions.

Four agents are concerned with the relation between d and s and in each case the amount of information that can be communicated from inference agent to acting agent is limited. These acting agents will be denoted by A_1, A_2, A_3 and A_4. It is also assumed that no acting constraints apply to any of these agents.

Agent A_1 knows prior to the inference what the experiment X is, and what the prior belief λ is. In this case, all the inference agent has to do is communicate experiment outcome x or some statistic of it. Such a "conclusion " would not be amplicative. As long as there is an efficient and flexible language for communicating statistics of x from inference agent to acting agent, amplicative inference will not be forced. There would be no advantage to selecting hypotheses or estimating parameters. If, instead, the language for communicating statistics of x from inference agent to acting agent is not efficient, then it is possible that some amplicative inference – e.g., some unobservable hypothesis relating d and s – might be a better option for A_1.

Agent A_2 knows prior to the inference what the experiment X is but not what the prior belief λ is. In this case, a non-amplicative inference requires that both λ (or some abbreviated version of it) and some statistic of x be communicated. If this can be done efficiently – within the brevity constraint – then there is no need for amplicative inference. If λ cannot be communicated efficiently then it is possible that selecting some amplicative inference might be a better option.

Agent A_3 knows prior to the inference what the experiment X is but not what the prior belief λ is. Agent A_3 does have its own prior belief about the relation between d and s and the inference agent trusts that it is close enough to λ. In this case, a non-amplicative inference requires only that some statistic of x be communicated.

Agent A_4 does not know prior to the inference what the experiment X is. In this case, for a non-amplicative inference to be communicated, the experiment itself (or some abbreviated version of it) must be communicated. If this cannot be done efficently within the brevity constraint then selecting some amplicative inference might be a better option.

To summarise: Constraints requiring conclusion brevity may – in the absence of other practical constraints – lead to situations where amplicative inferences are preferable. It may happen if the acting agent does not know – and cannot be told within the constraint – what the experiment/observation on which the inference was made is. It may happen if the acting agent has no appropriate prior belief about the phenomena of interest and cannot be given any within the constraint. It will not happen if the acting agent, prior to the inference: knows the experiment, has an appropriate prior belief, and has a sufficiently efficient and flexible method for receiving communications from the inference agent specifying statistics of experiment outcomes within the brevity constraint.

3.1.3 Motivations for Explanation

Using the conclusions from the previous subsection (3.1.2), it can be argued that conclusion brevity requirements are not a good motivation for explanation in formal/academic contexts. In those contexts the prior assumptions, experiments and observations must be communicated anyway; and given that, it follows that non-amplicative conclusions will be ideal regardless of intended conclusion application. In less formal contexts, where adequate descriptions of prior assumptions, experiments and observations are impractical, conclusion brevity requirements could be a valid motivation for amplicative inference.

Inference constraints are considered next. Remember that these are constraints on the function that is to be used for mapping premises to conclusions (see section 2.10). Realistically, there will always be such constraints; except in the most trivial cases where the set of premises and conclusions are both very small. To understand why, consider that the number of functions that can map premise set Λ to conclusion set Φ will be $|\Phi|^{|\Lambda|}$ – which will be uncountable if Λ is infinite. It is not realistic to search through all members of such large sets of candidate functions to find the best solutions. Despite this, it is actually easy to find one that will not be amplicative when the communication

constraint and conclusion language is relaxed enough to not force amplicative inference. All the inference algorithm has to do is repeat its input or some adequate descriptions of it. Assume that the communication constraint allows that the acting agent can be given adequate descriptions of the experiment X, the prior belief λ, and the experiment outcome x. In this case, all the inference function has to do is state those things, which are given to it as its input – little computation is needed. With this, we conclude that inference constraints, by themselves, are not good motivations for amplicative inference in formal/academic contexts.

Finally, acting constraints are considered. Acting constraints are limitations on how the agent that uses a conclusion can turn that conclusion into behaviour in its environment. If prior belief λ, experiment X and experiment outcome x are given to the acting agent as a conclusion, it may be the case that calculating the implications of that conclusion is computationally infeasible for the acting agent. For a limited agent to be capable of adopting some behaviour, it must have some set of states of mind that it can derive behaviours for. In general, there is no reason why there should be some such usable conclusion which exactly coincides with the ideal non-distorted premise – unless the prior belief was chosen specifically to lead to that property, as is the case with conjugate priors (see [Fink, 1997] for details on conjugate priors; see appendix B.2 for the relation between conjugate priors and amplicativity). If acting constraints rule out some non-amplicative inferences then they can force amplicative reasoning.

Consider now, inference in formal/academic contexts. It seems realistic to say that those who read and use the conclusions derived from analyses of data and experiments will have limitations in how they can compute the implications of those conclusions. By selecting a "best" hypothesis/explanation which allows easier calculation of degrees of belief in certain propositions, one may reduce the computational resources needed to use those conclusions in the environment of interest. This is the explanation that is often given for our tendency to uncover causes. Knowing the causal structure of a system simplifies the calculations needed to make predictions.

The conclusion of this section is now summarised: Explanation can be motivated by any of the three types of practical constraints (communication, inference and acting) but, in formal/academic contexts our model expects this effect to be due less to communication constraints and inference constraints and more due to acting constraints. Put

more simply, in formal/academic contexts, explanations are made primarily to reduce future computation. The desire to keep conclusions brief is secondary. Computational difficulty in making inferences (inference constraints) is secondary as non-ampicative inferences are easy to find; however, these constraints may compound the effects of acting constraints.

3.2 Occam's Selection Razor

Occam's razor advises preference for simpler explanations. This section will focus on how modern versions of it are used to justify inductive inference methods – as opposed to the original version. Unfortunately, in the field of inductive inference there are currently (at least) two separate concepts that are both being referred to as Occam's razor. One is concerned with the act of weighting hypotheses prior to making observations while the other is concerned with the act of selecting a best hypothesis after making observations. These will be refer to as the *weighting razor* and *selection razor* respectively.

The *weighting razor* is where simplicity of a hypothesis is used to justify giving it higher a prior weighting. This for example be seen in the algorithmic probability literature. Here each hypothesis is also a program. The simplicity of a hypothesis is the length of the encoding of its program for a chosen universal Turing machine. Solomonoff prediction ([Solomonoff, 1964, 1996]) makes predictions by weighting the contributions of these programs according to their program code lengths and is therefore said to use Occam's razor. Note that Solomonoff prediction does not explicitly select any single hypothesis – it rules out only those not consistent with the observation. The term "Occam's razor" is used in this context only to justify the method by which the hypotheses are weighted. The reasoning is as follows: Program code length measures simplicity of strings and simpler programs are given more weighting because of Occam's razor.

The *selection razor* can be seen in literature concerned with hypothesis selection and parameter estimation methods. Examples include the minimum message length (MML) and minimum description length (MDL) methods (see [Wallace and Dowe, 1999] for a discussion relating Occam's razor to MML and algorithmic probability). These methods also start with a priori weightings over hypotheses but do not necessarily use the term

"Occam's razor" to justify that weighting. Instead, simplicity is defined for hypotheses in light of both a priori weightings and the observations made. Here the "simplest" hypothesis is selected, in contrast to Solomonoff prediction where no hypothesis is selected.

It will now be argued that the second interpretation of Occam's razor (the selection razor) must be phrased in an interest-relative context. First, an example of the type of confusion that can occur when this context is not explicitly acknowledged is given.

From machine learning there have been several attempts to empirically discredit Occam's razor. These work by showing that more complex hypotheses often are better at prediction with real world data. Such empirical evidence has been provided by [Murphy and Pazzani, 1994] and [Webb, 1996] using decision trees.

In defence of Occam's razor Needham and Dowe [2001] reply with several arguments. Firstly, is was argued that using the number of nodes in a decision tree as the measure of hypothesis simplicity is incorrect and that the MML hypothesis coding-length formula is a better measure of hypothesis simplicity. Secondly, it was argued that hypothesis simplicity and goodness-of-fit must be weighed against each other quantitatively, which requires the use of decision trees using probabilistic predictions in their leaf nodes. Needham and Dowe [2001] fail to show empirically that the MML's preference for simple hypotheses necessarily maximises predictive performance.

We suggest that this approach to defending Occam's razor is invalid. MML is an inductive method for explanation and estimation, it is not a purely predictive method. To justify the preference for simplicity of estimation/explanation it is necessary to consider the weighting razor and selection razor separately.

First, it must be understood that use of the weighting razor logically implies that in most cases the best predictive conclusion will not be simple. If the data source actually conforms exactly to the a priori assumptions and data x was observed then the best predictive hypothesis will be the one θ that best imitates distribution $\Pr(y|x) \approx \Pr(y|\theta)$ where y is the variable to be predicted. There is no reason in general to believe that this θ will be simple by any measure. For decision trees it is very likely to be a tree that splits on all attributes along all paths. So, assigning high weight a priori to simple hypotheses in no way implies that simple hypotheses will be the best predictors. The weighting razor does not entail the selection razor.

To refute arguments against Occam's razor which use empirical evidence showing that best predictors are often not simple, it is necessary to think of the selection razor as a separate thing which must be phrased in an interest-relative context. There is no need for selecting hypotheses when there are no practical constraints. The selection razor is concerned with a different type of simplicity, one defined in terms of hypothesis usability (according to acting constraints) and communicability. If that type of simplicity is not desired for its own sake, if only predictive performance is desired, then the ideal conclusion does not need to be simple by any definition of simplicity.

In [Dowe, 2011, sec. 4] various incorrect arguments against Occam's razor are listed and refuted. It is also mentioned there that some arguments against Occam's razor rely on examples of MML and MDL coding schemes which are inefficient (in terms of two-part compression) and it is suggested that more efficient compression schemes would show Occam's razor in a better light. In section 3.5 we will show why that defence is misguided. The reason that practical MML methods select "simple" explanations is because interest information can enter the inference algorithm design via the choice of coding scheme employed (see section 3.5). If two-part compression were truly the only consideration, it would lead to inflexible estimators such as the strict minimum message length (SMML) estimator which does not in general select simple hypotheses. SMML will in general overestimate "order-of-magnitude parameters"; for example, it will almost always prefer giant decision trees even when a simpler tree defining a similar conditional likelihood function is available.

3.3 Measures of Inference Method Performance

New machine learning methods are constantly being developed and published. It is desirable to have objective methods for gauging their relative merits. For machine learning, predictive performance on validation sets are the usual measure. The data set is split into a training set and a validation set. First, the algorithm is trained on the training set. Finally, the conclusion's predictive value is measured by looking at its predictive performance on the validation set. In the terminology of the interest-relative model, the parameter values "learned" may be thought of as the conclusion that the algorithm produces while training is the act of inference.

When the interest-relative interpretation of inductive inference put forward here is adopted, there are two problems that this tradition of inference method validation can be expected to run into: Firstly, it becomes less meaningful to compare algorithms which project to different conclusion languages. Secondly, the measures by which predictive performance should be calculated for different algorithms might be irreconcilable.

To understand the first problem, imagine that two inference algorithms are to be compared empirically on some data set which has been divided into training set and validation set. Assume that the same measure of predictive performance on that validation set is appropriate for both. Algorithm i projects to hypothesis set Υ_i while algorithm j projects to hypothesis set Υ_j. If Υ_i is much more constrained that Υ_j then we can expect algorithm j to perform better on the validation set (see section 2.15). This, however, is because the two algorithms produce conclusions which are appropriate for different acting agents with different acting constraints. Agent i might, for example, be an agent that requires an explanation which explicitly identifies a single best causal structure while agent j allows a weighted set of multiple distinct causal structures as a conclusion. To attempt to compare the performance of i and j directly would wrongly give j preference while failing to acknowledge that i is concerned with a harder problem.

To understand the second problem, imagine now that algorithms i and j project to the same conclusion language, $\Upsilon_i = \Upsilon_j$. Assume that the intended environments for the two algorithms are different. These environments will be denoted by $\mathcal{E}_i$ and $\mathcal{E}_j$. If a measure of predictive performance is to be chosen for comparing inference algorithms it is desirable that it covers the types of predictions that are important to both environments $\mathcal{E}_i$ and $\mathcal{E}_j$. In the ideal situation one might combine $\mathcal{E}_i$ and $\mathcal{E}_j$ to produce some third "more invested" environment $\mathcal{E}_k$ where being adapted to $\mathcal{E}_k$ implies being adapted to both $\mathcal{E}_i$ and $\mathcal{E}_j$. In section 2.12 it was shown that such a more invested environment will exist when no practical constraints apply to the inference task. The measure of predictive performance could then be chosen according to what $\mathcal{E}_k$ values. A log-loss environment is what is approached when approximating the "most invested" environment (see appendix B.1). So, log-loss is the ideal when no practical constraints apply. In section 2.13 it was shown that when there are acting constraints, there might not exist any $\mathcal{E}_k$ which holds this property. There will be environments which are incomparable. As a consequence for machine learning, there cannot be a universal ideal measure of predictive performance

when acting constraints apply.

If the acting constraint is relaxed so that Υ_i contains more members, the chance that there exists some more invested environment $\mathcal{E}_k$ increases. Whether an ideal measure of predictive performance exists under which both algorithms i and j can be compared, depends on how severe the shared acting constraint is.

Log-loss is a popular measure of predictive performance. Log-loss is presented as a go-to method of predictive scoring as it is uniquely invariant to how the question about what is to be predicted is phrased [Dowe, 2011, sec. 3]. Our method of ordering environments (section 2.13) under constraints shows that there is no absolute best under practical constraints. In those cases, log-loss will still hold the property of being "maximal" (see the last paragraph of section 2.13). In other words, if log-loss represents the actual intended environment, there is no other environment that it can be replaced with that is strictly as, or more, invested.

In summary: While it is desirable that the performance of different machine learning algorithms be compared, fair comparison is not always possible. For two given algorithms, the degree of comparability will depend on how similar the acting constraints that motivate the two algorithms are. It will also depend on the intended environments in which the produced conclusion will be used. As acting constraints become more relaxed, the intended environments become less important and the possibility of a fair comparison method existing increases – the chance that log-loss will be appropriate increases.

In general then, there should not be an expectation to be able to fairly compare the performance of an algorithm intended for explanation with one intended for pure prediction. Even with two algorithms both concerned with explanation, they may be designed to produce very different types of explanations and thus still be incomparable.

There is a danger here for the practice of designing machine learning algorithms. Those who design the algorithms may feel pressured to incorrectly specify the interests behind specific algorithms in order to make them more comparable to existing methods and, to make them perform better on popular performance measures instead of a measure that captures what is actually desired.

3.4 The Parameter-Context Problem

Under deductive reasoning, when one believes proposition p, one must also believe all propositions entailed by it. If $p \Rightarrow q$ then q must also be believed. With inductive inference – under the interest-relative interpretation – such intuitions can be problematic.

Imagine that for a given observation x from experiment X a single hypothesis must be selected from hypothesis set Θ where $f(x|\theta)$ defines the meaning of hypothesis $\theta \in \Theta$. Assume that Θ is the crossproduct $\Theta = V \times W$. The prior probability of belief $\theta \in \Theta$ is denoted by $h(\theta)$. Imagine now that there are two acting agents, A_1 and A_2, concerned with the same environment. No inference constraint or communication constraint applies and the acting constraints are relaxed (see section 2.11 definition 17). For agent A_1 the conclusion language is (Θ, f). For agent A_2 the conclusion language is (V, g) where,

$$\forall x \in X,\ \forall v \in V,\ g(x|v) = \int_W f(x|(v,w))h(v,w)\ dw\ . \tag{3.1}$$

One can not, in general, expect that the best inference from Θ for A_1 will agree with the best inference from V for A_2 on what the value of v is; nor that their estimates will be close to each other when V is numeric.

This problem can in practice lead to a common pitfall when interpreting the results produced by inference algorithms. As an example, imagine that some Bayesian network was induced. Under the selected hypothesis (a completely instantiated Bayesian network), attribute x_{10} has a direct causal influence on attribute x_{51}. One might then be tempted to say that the inference method concluded that "x_{10} causes x_{51}", but that would be misleading. Here "x_{10} causes x_{51}" might only be a useful belief in the context of all the other things that the chosen complete conclusion says. By itself, it might be a terrible estimate.

For certain data models and processes this problem will go away as more and more data becomes available, but making the most of what data is actually available is also desirable. This *parameter-context problem* can be remedied to some extent if it is viewed as the consequence of a misspecified interest.

Return now to the first example used where A_1 needs an estimate from $V \times W$ while A_2 needs an estimate from V only. This can be related to the second example by

thinking of v as the statement "x_{10} causes x_{51}" while (v, w) is the complete Bayesian network instance that was estimated. If one reads "x_{10} causes x_{51}" from the estimate and uses that statement outside of the context of w then the interest of A_2 is adopted. If one uses the the entire estimate (v, w) then the interest of A_1 is adopted.

Let $\mathcal{E}_1$ and $\mathcal{E}_2$ denote the environments of A_1 and A_2 respectively. These conflicting interests could be reconciled by saying that an estimate must be chosen from $\Theta = V \times W$ which will be used in both environments $\mathcal{E}_1$ and $\mathcal{E}_2$. This could be achieved by creating a new hybrid environment $\mathcal{E}_3$. Environment $\mathcal{E}_3$ must be defined such that its expected utility for belief (v, w) must be equal to the expected utility of belief (v, w) under $\mathcal{E}_1$ times c plus the expected utility of belief v under $\mathcal{E}_2$ times $(1 - c)$. Here $c \in (0, 1)$ is a weighting constant which is needed because there might not be a single best way to combine $\mathcal{E}_1$ and $\mathcal{E}_2$. The behaviour set of $\mathcal{E}_3$ will be the cross-product of the behaviour sets of $\mathcal{E}_1$ and $\mathcal{E}_2$.

Imagine now that there is some new agent A_3 intended for $\mathcal{E}_3$ which takes estimates of the form (v, w) as conclusions. Since the behaviour for the $\mathcal{E}_2$ part of $\mathcal{E}_3$ must be optimised in a way that ignores the estimated value of w, this hybrid agent might need to be defined using an acting constraint that is not separable (see definition 15 section 2.11) even if the two original acting constraints were both separable.

Note that even if the interest is redefined in this way to reduce the parameter-out-of-context problem for V from $\Theta = V \times W$, there are still many other ways to re-parametrise Θ. Addressing the problem for one parameter will not necessarily fix it for others. If one defines an environment which combines too many such parameter-out-of-context environments, the result may be an unacceptable level of compromise.

The class of separable acting constraints was defined in section 2.11 to represent a range of "reasonable" acting constraints; the class for which the set of behaviours that may be considered does not depend too much on what conclusion is believed. In this section it can be seen that even that class (the separable class) might be too ideal to match the types of interests for which the parameter-context problem is a concern. The parameter-context problem is not uncommon for inference methods intended for explanation – some examples will be given for MML in section 3.5. This suggests that the acting constraints which are implicit in common machine learning methods could be quite complicated (not separable).

3.5 The Minimum Message Length Principle

The minimum message length (MML) [Wallace and Boulton, 1968, Wallace and Freeman, 1987, Wallace, 2005] and minimum description length (MDL) [Rissanen, 1978, Grünwald, 2007] principles define two popular and similar paradigms for guiding the design of inductive inference algorithms. These are arguably two of the most widely applicable principles for that purpose. Both define explanation using data compression. We will focus here on MML as it is more explicitly a Bayesian method, making it is easier to apply our model.

One prediction of the interest-relative interpretation of explanation is that any principle/paradigm that guides the design of inference algorithms will perform poorly in selecting explanations if it does not allow information about interest to enter the problem formulation.

Luckily, the principle/paradigm does not need to explicitly be defined in terms of interest. As long as there are many solutions allowed for a given problem, a practitioner may choose one that suits their interest better. This will be demonstrated by comparing strict minimum message length (SMML) inference with the more practical minimum message length (MML) inference.

SMML has a very exact definition allowing little interest information to enter the formulation. While it is not computationally feasible, even if it were, its estimates can be expected to have some properties that are undesirable for explanations (to be described in subsection 3.5.1). MML is a more practical approach to inductive inference. The practitioner has more flexibility with constructing MML inference methods. That allows for more interest information to enter the formulation – even though MML is not presented in the literature with interest as an explicitly identified ingredient. Subsection 3.5.2 describes how MML allows this to be achieved, why it is beneficial, and suggests how that knowledge may be used to make the application of MML easier.

3.5.1 Strict Minimum Message Length Inference

This subsection looks at how interest can enter the application of SMML. SMML, like MML, is not explicitly presented as interest-relative in the literature. Utilities and

environments do not enter its formulations. SMML is concerned with defining a best explanation for a given observation and prior belief. SMML was introduced by Boulton and Wallace [1975] and is described in [Wallace, 2005, chap. 3]. More recent work in SMML includes [Dowty, 2014] and [Dowty, 2015].

An experiment X will be performed where X is discrete. First, a set of hypotheses Θ is assumed. Each hypothesis $\theta \in \Theta$ defines a distribution over the members of X where the probability of outcome x given hypothesis θ is defined by $f(x|\theta)$. Here, $f(x|\theta)$ is called the likelihood of θ given x. A prior probability measure h is assumed over the set of hypotheses Θ. The marginal distribution over X is denoted by r and is defined as,

$$r(x) = \int_{\Theta} h(\theta) f(x|\theta) \, d\theta \, . \tag{3.2}$$

For a given outcome $x \in X$, SMML selects the hypothesis which leads to an optimal two-part encoding of x when using a specific type of encoding scheme. The encodings are made up of two parts: The first part is known as the *assertion* and it specifies a single hypothesis from Θ. The second part is known as the *detail* and it specifies the observed x using a coding scheme that is specific to the hypothesis selected by the first part.

The codebook assigns "codes" to some subset of Θ. The code length of the assertion (first part) code for hypothesis $\theta \in \Theta$ is denoted by $L_s(\theta)$ where s is some codebook. The length of the detail (second part) code for $x \in X$ given $\theta \in \Theta$ is denoted by $L(x|\theta)$. The coding scheme for the second part is designed to be optimal in expectation when the selected hypothesis is true. This leads to the equation $L(x|\theta) = -\log f(x|\theta)$. The codebook s for the first part is designed to minimise the expected length of the two-part code-length sum, $L_s(\theta) + L(x|\theta)$. Here, the expectation is taken over the marginal distribution $r(X)$. To find the optimal SMML codebook s, the probability function $s'(\theta) = e^{-L_s(\theta)}$ that minimises this two-part expected encoding must be found. Note that the optimal codebook s will assign non-zero probability to only a discrete subset of Θ as the codes must be finite to be decodable. The optimal SMML estimate for observation x will be denoted by $m(x) \in \Theta$.

Under SMML, $m(x)$ is the best "explanation" for observation x. In practice, SMML is not used because – amongst other reasons – it is computationally infeasible to find the

optimal codebook s. The term MML is used to refer various practical approximations to SMML.

Now that SMML has been described, consideration is given to what goes into it and where interest fits. For a best SMML explanation to be defined one needs to first define: the experiment X, the set of hypotheses Θ and their meanings f, and a prior probability h over Θ. There is nothing here directly corresponding to environments or practical constraints.

In the SMML literature, (Θ, f) is presented as prior belief. The set of unobservables Θ that are believed to exist a priori is part of that prior belief. This interpretation is incompatible with our assumption from section 1.23 where it was assumed that: Assumption A, "the value and meaning of a belief is derived only from how it influences the actions of agents in environments". Assumption D, "unobservable entities are only relevant to environments through observable entities. If all observables are known, there is no value in learning unobservables". For our model, the marginal distribution $r(X)$ captures all prior belief. Unobservable hypotheses are only useful tools.

For SMML to be considered under the interest-relative interpretation of induction, we suggest that the choice of conclusion language, the pair (Θ, f), should be seen as the formulation of a practical constraint. Under this interpretation, SMML defines the best explanation for a given observation x relative to a prior belief r and conclusion language (Θ, f). One consequence of this is that it need not be possible to define some $h(\cdot)$ over Θ which leads to marginal distribution r.

One counter-argument that might be raised to this suggestion is that the belief about what the unobservable entities are, is part of the prior belief. In other words, the specific parametrised model may be considered essential to the prior belief and explanations must concern its unobservable entities. This would contradict the assumption that we have made in section 1.23 that the value and meaning of a belief is derived only from how it influences the actions of agents in environments, but our assumption should hardly be law to MML users.

Our argument for why (Θ, f) should be seen as the formulation of a practical constraint is that the SMML mechanism of explanation depends on properties of the data model (Θ, f) which seem arbitrary under the interpretation of Θ as prior belief but not under the interest-relative interpretation. This will be demonstrated by considering two

extreme cases which we will refer to as *under-explanation* and *over-explanation*. By considering these extremes, it will become clear that the ability of SMML to select conclusions that have the properties of explanations, or practical predictive tools, depends on how the conclusion language is restricted and *that* it is restricted.

The aim in the remainder of this subsection will be to demonstrate some problems that could arise if one were to stubbornly attempt to apply SMML according to an interest-objective ideal. These examples are not representative of the intended use of SMML.

Under-Explanation:

SMML is described as dividing the information contained in observation x into pattern information (contained in the assertion) and noise information (contained in the detail). The amount of information $L_s(\theta)$ contained in assertion θ is a measure of how much the inference "explains".

Under-explanation is where the amount of explaining that the inference method does decreases as the conclusion language becomes more expressive. Imagine that one is allowed to re-parametrise hypothesis language (Θ, f) by adding a new hypothesis μ to the conclusion language thus creating $\Theta' = \Theta \cup \{\mu\}$. Imagine also that h and f are replaced with h' and f' where $\forall \theta \in \Theta,\ f'(\cdot|\theta) = f(\cdot|\theta)$ while h' is defined such that the marginal distribution r is unchanged.

By selecting μ a certain way it is possible to reduce the expected message length $L_s(\theta) + L(x|\theta)$ for the optimal codebook while also reducing the expected assertion length $L_s(\theta)$. Let θ_1 and θ_2 be two possible assertions for the original parametrisation – so $L_s(\theta)$ is defined and finite for both. Now define μ such that,

$$\forall x \in X,\ f(x|\mu) = e^{-L_s(\theta_1)} f(x|\theta_1) + e^{-L_s(\theta_2)} f(x|\theta_2)\ . \tag{3.3}$$

For the new parametrisation (Θ', f'), a new optimal SMML codebook s' can be made which is more efficient than s. Remove the codes for θ_1 and θ_2 and replace them with a single code for μ.

As an extreme case, imagine that the hypothesis added is λ where $\forall x \in X,\ \lambda(x) = r(x)$. Let $h'(\lambda) = c$ while $\forall \theta \in \Theta,\ h'(\theta) = h(\theta)(1-c)$ where $c \in (0,1)$. This leads to r being unchanged. Now the optimal SMML codebook can be constructed by having only a single code – one corresponding to λ. The resulting two-part message will decrease in expectation while reducing the assertion length to 0 for all possible observations.

It can be seen here that making the hypothesis set more expressive can lead to shorter assertion lengths. In the extreme case that assertion will not depend on the observation at all and the inference could not be said to do any explaining at all.

For those problems that MML is typically applied to, the data x can be divided into items $x = (x_1, x_2, \ldots, x_n)$ where the items are conditionally independent for any given hypothesis, $f(x_i \wedge x_j|\theta) = f(x_i|\theta)f(x_j|\theta)$. With this condition the under-explanation problem will a least have a lower limit on the expected assertion length.

It has now been described how SMML will explain less as the hypothesis set considered becomes less restricted; even when the resulting a priori marginal distribution is unchanged. For the case where the data can be divided into items which are believed a priori to be conditionally independent on some unobservable parameter, there seems to be some degree of objectivity that can be achieved in deciding how the hypothesis set should be limited. In the more general case, it is less clear what is to stop one from making hypothesis language more and more expressive until SMML does no explaining at all.

If the interest-relative interpretation of induction is adopted then the possibility of under-explanation is an expected feature of any inductive method that allows interest to enter the problem formulation. An expressive conclusion language implies weak practical constraints which means little explaining is needed.

Over-Explanation:

The reverse of under-explanation is over-explanation. The reader should be careful not to confuse over-explanation – a term introduced to aid the discussion here – with over-fitting.

The SMML estimator $m : X \to \Theta$ partitions the data variable X into coding groups. For a given group, m maps all its members to the same hypothesis. Denote the coding

group for $\theta \in \Theta$ here by $t_\theta = \{\ x \in X \mid m(x) = \theta\ \}$. Imagine now that the hypothesis space may again be re-parametrised by adding new hypotheses to it. Some t_θ which has multiple members can then be taken and divided into two; call them t_1 and t_2. Now define two new hypotheses, μ_1 and μ_2, where each imitates θ over its respective subgroup while assigning much smaller probabilities to all other members of X. In this way, the hypothesis set can be extended to include hypotheses which are more specific. The expected two-part code length will decrease while the expected assertion length will increase.

As with under-explanation, the over-explanation will have an upper limit (in assertion size) if the data can be divided into items where the hypotheses must make all items conditionally independent of each other.

As an extreme case, a single hypothesis μ_x could be defined for each possible observation $x \in X$ where $f(x|\mu_x) = 1$. This would lead to optimal compression while assigning all information to the assertion. This extreme case becomes obvious when attempting to apply SMML directly to a Solomonoff prior. Here each program p is assigned a weighting equal to $2^{-|p|}$ where $|p|$ is the code length of the program according to a chose universal prefix Turing machine. The marginal probability of finite binary string x being observed is then $\sum_{p \in P_x} 2^{-|p|}$, where P_x is the set of all programs that halt and output x. Note that this is not a proper probability function as the sum of probabilities over all X does not add to 1; since many programs will not halt. If this model of the set of all finite binary strings were taken as prior belief and SMML were applied directly, it would assign a single hypothesis (program) to each member of X (the set of all finite binary strings). Such inferences would obviously not be useful for prediction, nor as explanations. In [Wallace and Dowe, 1999, sec. 7] this is noted and a method is suggested by which programs form a Solomonoff prior might be grouped together to form more general hypotheses. It is also stated there that in scientific enquiry there is a desire to find specific theories from which deductions can be easily made.

The Assumed Metaphysics:

From these examples, it can be seen that the amount of pattern information that SMML sees, depends on types of hypotheses that are contained in the assumed data model. A model that involves many unobservable entities will force SMML to explain more while

leading to worse prediction. A model that involves few unobservable entities will lead to little explaining but better prediction. This is assuming that both models lead to the same marginal distribution r (see equation 3.2).

Under the interpretation of the conclusion language as being determined by prior belief, this would not be considered a problem since the conclusion language would not be determined exactly by the data model language. An example is now given to illustrate how awkward inference would become if prior belief literally determined which unobservable entities are estimated. Imagine there is an object. Initially its position is known. Every discrete time interval it will move either one unit of distance to the left, or one unit to the right. The probability of moving left is p. Its position is measured at the end of each cycle of 100 discrete time intervals. Let x represent the series of positions observed at the end of each 100 discrete time intervals. Let z represent the unobserved series of all left/right movements. Here, according to the prior belief, the unobserved entities are p and z. If SMML were asked to infer estimates of all unobserved entities given x, it could give an exact estimate to both p and z as hypothesis. The combination of all left/right movements y determines x exactly. Even worse, once y is estimated, the estimated value for p could be anything since x is conditionally independent of p given y.

A more reasonable application of SMML would simply treat this problem as a binomial problem; to estimate p and not attempt to estimate y. This example was given to demonstrate how a common sense solution could be missed if one dogmatically attempted to estimate according to the actual prior belief held.

The examples given should highlight the general problem: As the metaphysical model constructed becomes more detailed, attempts to estimate all unobservable entities will lead to over-explanation. Conversely, if an expressive range of highly stochastic hypotheses are included in the data model, one may encounter the under-explanation problem.

If the interest-relative interpretation of induction is adopted, the unobservable entities that are employed by the data model expressing one's prior belief, need not be the unobservable entities that will be estimated. The degree of explaining will depend on the interest and not on how detailed the a priori model formulation is.

The SMML Environment:

There does not seem to be anything in MML or SMML theory that can be directly related to an environment. From the term $L_\theta(x|\theta) = -\log f(x|\theta)$, which goes directly into the term that SMML minimises, $L_s(\theta) + L_\theta(x|\theta)$, it can be seen that SMML uses a log-loss measure of goodness-of-fit, $-\log f(x|\theta)$. If, in the language of the interest-relative model, there is some specific environment of interest $\mathcal{E}$, there is not much that can be done to tell SMML about it. As was discussed in section 2.12, even two environments with the exact same sufficient statistic may differ in how they value its information and that becomes relevant when practical constraints are present.

The log-loss measure $-\log f(x|\theta)$ values all information produced by experiment X equally. If the environment of interest $\mathcal{E}$ does not value all information equally then it should be expected that an inference methods using log-loss will become less ideal as the practical constraint becomes more severe. The problem can be avoided by choosing a hypothesis language which is sufficiently flexible; but that may conflict with the actual practical constraint. As noted in section 3.4, the parameter-context problem demonstrates how failing to identify less ideal aspects of the environment of interest can lead to problems.

3.5.2 Practical Minimum Message Length Inference

Strict minimum message length estimators are not used in practice. SMML codebooks are usually computationally infeasible. The focus within the preceding SMML subsection was not to demonstrate how SMML is actually intended to be used, but to demonstrate the problems that could arise if one were to stubbornly attempt to apply it according to an interest-objective ideal.

In practice, approximations to SMML are used. These form what is known as minimum message length (MML) inference. These approximation methods often avoid the problems mentioned in section 3.5.1. It will now be argued that one of the reasons that certain MML coding schemes in practice often avoid the parameter context-problem (section 3.4), or the under-explanation problem, or the over-explanation problem, is that they allow for interest information to enter the data compression scheme design.

SMML works by partitioning the observation space X such that each subset defines a *coding-group* for which the estimated hypothesis $m(x)$ is the same. Denote the coding group for $\theta \in \Theta$ here by $t_\theta = \{\ x \in X \mid m(x) = \theta\ \}$. The coding length for assertion θ is then $\sum_{x \in t_\theta} r(x)$, the marginal probability of partition t_θ.

Most practical MML algorithms use approximations to an imagined coding scheme where the coding-groups correspond to subsets of the hypothesis space Θ. Let $R_x \subseteq \Theta$ denote the coding-group that will be selected for observation $x \in X$. The group R_x will have a representative $m(x) \in R_x$ which is used as the inferred estimate for x. The code length for asserting that representative will be $L(m(x)) = \int_{R_x} h(\theta)\ d\theta$. The length of asserting the detail is again the negative log likelihood $L_{m(x)}(x|m(x)) = -\log f(x|m(x))$. It can be imagined that these code-groups partition the space Θ in a way that leads to the two-part code length $L(m(x)) + L_{m(x)}(x|m(x))$ being minimised in expectation over $r(X)$. In practice, this partition is not explicitly calculated and the code-group sizes and representatives are calculated and selected by approximations. The details of those approximations will be omitted here as they are somewhat involved and not essential to our discussion. See [Wallace and Freeman, 1987] and [Wallace, 2005, chaps. 4–5] for details on this type of approximation.

A regression problem will now be used to show how this method of constructing coding schemes can be used to avoid the parameter context problem. Let each possible observation $x \in X$ consist of a set of items where each item is a two-dimensional point in some bounded rectangle. The first attribute of each item is an independent variable while the second is a dependant. Polynomial regression will be used to model the relation between the first and second attribute. The order of the polynomial is denoted by $k \in \{1, 2, \dots\}$. The remaining parameters will be denoted by σ. Each hypothesis will then take the form $\theta = (k, \sigma) \in \Theta$. Note that the dimension of σ will depend on k. For $k = 1$ the polynomial will be a straight line; for $k = 2$ it will be a quadratic.

SMML can not be expected to estimate the polynomial order k well for this problem. If, in truth, the data is generated by a straight line, i.e. $k = 1$, the selected SMML estimate is likely to be a polynomial of the highest order available which has a shape very close to a straight line inside the rectangle containing the data (see, e.g., [Dowe, 2008, footnote 153]). SMML does not know that a code-group representative of lower order would be preferred.

An MML coding scheme for this problem will code the assertion in two separate parts. First it will specify what value of k is estimated using a code not dependant on the estimated σ, then it will state σ using a code specific to the estimated value of k. The coding length of estimate $k = 3$ will then simply be $-\log h(k = 3)$, the negative log likelihood of the prior probability of k being 3. See [Wallace, 1997] and [Fitzgibbon et al., 2002] for examples of such MML coding schemes for polynomial fitting.

This solution prevents any coding-group $R \subseteq \Theta$ from containing members with different estimated values of k. The result is that a high order polynomial which is very close to some lower order polynomial in shape will never be selected. Note, this "close-in-shape" property will depend on how much data is available. Here the MML is likely to select the correct polynomial order or, when too little data is available, one of lower order (see [Dowe, 2008, footnote 153]).

The difference in SMML and MML estimation described here is not due to how different polynomial orders are weighted a priori. SMML will overestimate polynomial order even when low-order polynomials are significantly favored a priori. The MML solution is unlikely to overestimate the polynomial order and this does not require a prior heavily skewed in favor of lower order polynomials.

Next, a decision tree example is considered. For simplicity, the tree will have only binary attributes to split on at the nodes. Let a_j represent the jth attribute where $j \in \{1, 2, \ldots, M\}$. Let $x_{ij} \in \{0, 1\}$ represent the value of attribute j in data item i. Let $y_i \in \{0, 1\}$ represent the classification of item i. The goal here is to find a decision tree which represents the conditional relation of y given x. Each tree $\theta \in \Theta$ will be made up of two components $\theta = (k, \sigma)$; k represents its structure while σ represents the leaf attributes. The prior distribution h is defined such that $h(k_1) < h(k_2)$ when k_2 has more internal nodes.

For SMML, the estimated trees will tend to be much larger than the true tree. Each code group representative will likely describe a distribution of y given x which is very similar to some much smaller tree. This is an instance of the parameter-context problem. SMML does not know that it is desirable that the shape of the tree may be easily interpreted outside of the context of the estimated leaf parameters.

Let's look at how a typical MML coding scheme would differ. If a way is sought to partition Θ into code-groups without any restrictions on which members of Θ may

be grouped together, then there is no reason why the code-group representatives would resemble the majority code-group members in the desired aspect – in this case, tree structure and size.

A more desirable MML coding scheme for binary decision trees is now described. The assertion will be stated in two distinct parts: first the tree structure is stated, then the leaf class proportions are stated. Different tree structures are put in the same coding group iff there is a one-to-one correspondence between their leafs. Two leafs can correspond only if the paths leading to them split on the same set of attributes with the order of those attributes being irrelevant. For this coding scheme, a large tree θ_1 which closely resembles some smaller tree θ_2 – resembles in the sense that $f(\cdot|\theta_1) \approx f(\cdot|\theta_2)$ – will not be selected. Again, similarity $\approx$ depends on data set size. The subset of Θ with members similar to θ_1 which also have structure equivalent to it will have smaller a priori probability than the subset of Θ with members similar to θ_2 which have structure equivalent to it. Examples of such MML decision tree coding schemes can be found in [Wallace and Patrick, 1993] and [Tan and Dowe, 2004].

As a final example, the MML estimator for continuous parameters (see [Wallace and Freeman, 1987] and [Wallace, 2005, chap. 5]) is considered. Assume that Θ is some continuous space. Here the coding-groups will be (hyper-)spheres in parameter space under an orthogonal re-parametrisation of the parameter space. The re-parametrisation is determined by a local quadratic approximation to the likelihood function. Imagine that θ_1 and θ_2 have similar likelihood functions $f(\cdot|\theta_1) \approx f(\cdot|\theta_2)$. This MML estimator will not put θ_2 in a group with representative θ_1 if there is some region in space Θ separating θ_1 and θ_2 for which the members do not resemble θ_1 sufficiently in likelihood function behaviour. This continuous parameter MML estimator does something similar to the discrete parameter examples given above. It puts restrictions on how hypothesis which are "structurally" different may be put in the same code group.

The examples of MML coding schemes that avoid the parameter context problem presented above all have one thing in common: they all avoid the problem by restricting which hypotheses may be put together in coding-groups. If such restrictions are essential to constructing useful MML coding schemes, one must ask, what principle guides how it should be done? It could be suggested that the metaphysics of the a priori data model might be used to guide it. That possibility will now be considered and then argued

against.

Looking at the polynomial regression example, it could be suggested that all estimates of discrete parameters must be exact; i.e., hypotheses that give different estimates for some discrete parameter must not be put in the same coding group. That suggestion is easily countered by noting that MML coding schemes do often put hypotheses with contradicting estimates for discrete parameters in the same coding-groups and that failing to do that can lead to very inefficient compression and biased estimators. A good example is MML unsupervised classification problems (see [Wallace and Dowe, 1994], [Wallace and Dowe, 2000], [Visser and Dowe, 2007] and [Visser et al., 2009b]). There, a number of observed items must be assigned to different classes. The assignment of things to classes in the orthodox MML solution is not exact. The coding groups are imagined in a way that groups together hypotheses which disagree on which things belong to which classes. If this is not done, the class parameters would be badly estimated. Such imprecise coding schemes are described in [Wallace, 2005, sec. 6.8] and are sometimes used for more complex discrete parameters (e.g. [Wallace, 1998] and [Visser et al., 2009a]).

Another suggestion that might be inspired from the polynomial regression example is that one should never group together hypotheses which disagree on estimates of "order of magnitude parameters". In the polynomial regression example, the parameter k specifies the polynomial order and is an order of magnitude parameter. For an unsupervised classification, the parameter stating the number of classes is an order of magnitude parameter. One problem here is that for many more complicated hypothesis spaces, it might not be obvious what the order of magnitude parameters are; consider Bayesian networks and decision trees. This particular principle would lack generality.

It will now be argued that when the interest-relative interpretation is adopted, the types of coding group restrictions demonstrated in this subsection can be justified and guided by interest. SMML allows one to specify some information about the practical constraint through the choice of conclusion language but does not allow any information about the intended environment to enter the problem specification. Using MML coding group restrictions, information about the intended environment can enter the problem specification.

If a parameter will be taken out of context in the environment (see section 3.4 for the parameter-context problem), estimates which disagree significantly on their value

for that parameter can be prevented from being placed in the same coding group. For an order of magnitude parameter, that might be achieved simply by requiring that all hypotheses in the same group must agree exactly on its value. For parameters describing structure, such as decision tree structure, effective equivalence or similarity in degree could be required – as is done with Bayesian network structure (see [Comley and Dowe, 2003], [Comley and Dowe, 2005], [Wallace, 2005, sec. 7.4] and [Visser et al., 2012] for examples of MML applied Bayesian networks). For continuous parameters, similarity can be required by restricting the shape of coding groups – as is done with the I1B estimator [Wallace, 2005, chap. 5]. Simple ball-like shapes might be preferred over more complex Swiss-cheese-like shapes.

If the conclusion language is too flexible – leading to under-explaining inferences – the coding group restrictions may be defined such that hypotheses which are very different in terms of how they will be acted on will not be grouped together. Here a very large decision tree might not be grouped together with a small one because they would not be used in a similar way in the intended environment even if their predictive distributions are very similar.

There seems to be no apparent way to use coding group restrictions to remedy the over-explaining problem. That problem can be fixed by extending the conclusion language. For example, by appending to the conclusion language weighed unions of existing hypotheses.

To conclude: If one attempts to interpret MML as interest-absolute, the parameter-context problem, the under-explanation problem and the over-explanation problem all seem like anomalous side-effects. An interest-absolute interpretation cannot explain why, in practice, MML tends to avoid these problems. The interest-relative interpretation can explain it. These problems are avoided when the conclusion language matches one's practical constraint and when the coding schemes are designed with an awareness of the intended environment. For these reasons, we suggest that a more openly interest-relative interpretation of MML would be more useful than what is traditional for MML. Chapter 4 will demonstrate.

We suggest that in general, any method or principle that defines what the best explanation for an observation is, without allowing interest information to enter the problem specification, will either only be applicable to a very restricted class of problems,

or will for many data source models produce conclusions which do not have the properties desirable for explanations.

Chapter 4

An Interest-Guided Approach

4.1 A Hybrid-Agent Method

In the preceding chapter (3), it was argued that if some approach to inference algorithm design is meant for producing explanations then it must somehow allow interest information to enter the problem specification. This is still a somewhat abstract notion so in this chapter the concept is demonstrated using decision trees as examples. The goal here is to demonstrate a practical interest-guided approach to inductive inference algorithm design.

The concepts of practical constraints and environments were useful for describing the relation between interest and induction; but, they are perhaps too flexible to be directly useful for guiding algorithm design. We now suggest some principles for practical interest specification which will be placed under the label *the hybrid-agent method*.

The core idea here is that a complex interest might be more easily specified if it were described as a compromise between multiple simpler agents, each having a more pure and easily described interest. One starts by designing separate inference algorithms for each agent. After that, those solutions are merged into a hybrid algorithm. We suggest two principles to follow for achieving such hybrids: *parallel interpretations* and *mixed objectives*.

Parallel interpretations is a notion to be used when two simple agents require separate conclusion languages. For example, imagine that agent A_1 requires an estimate

of parameter Θ_1 while agent A_2 requires an estimate of parameter Θ_2 where $\Theta_1 \neq \Theta_2$. Here, the solution is to estimate some parameter Θ_3 which determines both Θ_1 and Θ_2. One then imagines that A_1 will take an estimate of Θ_3 and ignore the part not contained in Θ_1. In other words, the simpler agents might interpret a given estimate $\theta \in \Theta_3$ as defining different probability functions.

Mixed objectives is a notion to be used once a conclusion language has been selected for the hybrid agent. Let $G_1(\theta)$ be the objective that agent A_1 would prefer to optimise when selecting from set Θ_1. Let $G_2(\theta)$ be the objective that agent A_2 would prefer to optimise when selecting from set Θ_2. Mixing criteria simply means finding a hybrid objective G_3 that weighs G_1 and G_2 against each other of in some way. Here it helps if the units that G_1 and G_2 are measured in are the same and measure something similar.

4.2 A Bayesian Decision Tree Model

A decision tree problem will be used to demonstrate application of the hybrid-agent method. It is assumed that the reader is familiar with decision trees (if not, see [Wallace and Patrick, 1993] as an easy introduction).

Let $z = (x, y)$ be the observed data set where vectors $x = \{x_1, x_2, \ldots, x_N\}$ and $y = \{y_1, y_2, \ldots, y_N\}$ represent N observed data items. Each $x_i = \{x_{i,1}, x_{i,2}, \ldots, x_{i,M}\}$ is a vector of attributes for item i while each $y_i \in \{1, 2, \ldots, K\}$ is a classification for item i. Decision tree models are used in supervised classification problems where the goal is to predict the classifications of future items for which the attribute values are known.

The a priori assumptions of the decision tree model will now be described. These assumptions are both the a priori assumptions used to derive the inference methods to follow, and are to be used to generate artificial data for evaluating those methods.

It will be assumed that attribute values are independent within each item, i.e., $\Pr(x_i) = \prod_{i=1}^{M} \Pr(x_{i,j})$. This is not a good assumption for real data, but it makes the mathematics a lot simpler when implementing solutions that minimise expected Kullback-Leibler divergence. It is hoped that this simplification does not detract too much from the concepts that this chapter is meant to illustrate.

If q is the number of times attribute j had a particular value in the training set x then $(q + 0.5)/(N + 0.5c_j)$ will be used as the assumed probability of that particular value for attribute j occurring in future data items (e.g. in the validation set – since actually looking at the validation set proportions would be cheating). c_j denotes the number of values that attribute j may take.

Let Θ represent the set of all decision trees applicable to the given data set. A prior distribution $h(\Theta)$ will be defined. Let p_d represent the prior probability that a given node at depth d is a split and not a leaf – so the probability of the root node being a split is p_1. Let the probability of splitting on attribute $j \in \{1, 2, \ldots, M\}$ be equal for all attributes on all splitting nodes.

All splits will lead to 2 child nodes; the method for splitting on non-binary (possibly scalar) attributes must always give two outcomes. For categorical attributes with more than 2 possible values, the split node must specify a single value that leads to the right child node while all other values lead to the left child node. For integer or scalar valued attributes the split node must specify a cut-off value where values less than it lead to the right child node while all other values lead to the left child node. The possible cut-off value will be limited according to the distribution of the values of that attribute in the training set.

The leaf parameters (associated with each leaf of a tree) will be multinomial distributions describing the probabilities of each class appearing. The a priori distribution over these leaf parameters will be a uniform multinomial Dirichlet distribution with alpha value $\alpha = 0.7$. The alpha value describes an a priori belief about how 'extreme' the multinomial distributions are expected to be. With $\alpha < 1$, higher weight is given to multinomials with probability vectors lying on the edge of the simplex (as one would expect from informative leaves). $\alpha = 0.7$ was chosen for that reason but the exact value is somewhat arbitrary.

The tree model to be used will differ from standard practice by allowing identical splits (non-informative splits) to occur multiple times along a single path. Some of our algorithms work by sampling from the posterior distribution $\Pr(\Theta|z)$ where Θ is the set of all trees. The Metropolis-Hastings algorithm is used for this sampling. Allowing redundant splits allows for a faster tree sampling algorithm. See appendix C.1 for a more detailed description of the sampling method used.

4.3 A Predictive Agent

Under the hybrid-agent method, one starts by designing inference algorithms for simpler agents with more pure interests. Our first simple agent will seek to maximises predictive performance for the next single observation according to the expected log-loss measure. This agent does not care what form the estimate takes – anything that the computer can handle will do.

Log-loss is not "the" universal ideal for measuring predictive loss when practical constraints apply (as was argued in section 3.3), but it is commonly used and has some unique merits (see [Dowe, 2008, footnote 175], [Dowe, 2011, sec. 3] and [Dowe, 2013, sec. 4.1]). When no practical constraints apply (or very few, as is assumed for this agent), log-loss can be seen as the ideal – assuming that optimising expected loss is preferred (see appendix B.1). Minimising expected loss, as opposed to some other quantity (e.g. worst case loss) is not always ideal but it can be argued for in that it will give the highest average utility if the process (inferring decision forests) is repeated indefinitely.

Let $z' = (x_{N+1}, y_{N+1})$ denote the next item for which predictive performance must be optimised. The goal is to find a conclusion ϕ for which $\phi(y_{N+1}|x_{N+1}) \approx \Pr(y_{N+1}|z, x_{N+1})$. Note that the attribute vector x_{N+1} is not known when a conclusion is to be selected.

Since minimising expected log-loss of the prediction is desired by this agent, minimum expected Kullback-Leibler divergence (MEKLD, or minEKL) is the objective that should be optimised for. Since there is little constraint on the form of the desired conclusion, a decision forest can be used (multiple trees as a single estimate). Let Θ denote the space of all fully instantiated trees (with both structure and leaf node parameters specified). The conclusion language is therefore $\Phi = \Theta^T$ where T is the number of trees in the forest. Predictions are made using,

$$\phi(y_{N+1}|x_{N+1}) = \frac{1}{T} \sum_{\theta \in \phi} \Pr(y_{N+1}|\theta, x_{N+1}) \ . \tag{4.1}$$

Our solution samples T trees from the probability distribution $\Pr(\Theta|z)$ and uses that as the conclusion. The Metropolis-Hastings algorithm is used for this sampling. Note, we do not sample the leaf node parameters randomly as that would be too computationally expensive. Using a Dirichlet prior (being a conjugate prior) over leaf node parameters

allows estimation without amplicativity (see appendix B.2) – i.e., parameterless estimation (see [Fink, 1997] for details on conjugate priors).

While MEKLD estimates for this predictive agent can be expected to perform well according to predictive performance, they will not work well as "human" explanations because trying to comprehend what a collection of thousands of trees says about the underlying mechanisms of the phenomenon being studied is not directly possible. This leads us to the next simple agent.

4.4 The MML-Agent

To find an "explaining" estimate, it makes more sense to look for a single tree estimate which is probably similar in structure to what the "true" tree looks like. Minimum message length (MML) decision trees can be used to produced such estimates. MML will produce trees which balance goodness-of-fit (according to log-likelihood) against tree complexity. This method will not produce trees larger than what is justified by the training set and will seldom overestimate tree size when using artificial data from the assumed prior distribution (see section 4.6).

Our implementation for this MML-agent follows the method from [Wallace and Patrick, 1993]. Note that there are more developed versions of MML decision tree estimators (see [Tan and Dowe, 2004]) but we will follow the simpler version to make our hybrid-agent algorithm design easier.

The code length for stating that a node is a leaf is $-\log(p_d)$ where d is the depth of the node. The vector $p = (p_1, p_2, \dots)$ is defined a priori. The code length for stating that the node is a split on attribute j is $-\log(1-p_d)+\log(M)+\log(c_j)$ where c_j denotes the number of values that attribute j might take. The $\log(c_j)$ term may is dropped when $c_j = 2$. Remember that M is the number of attributes and that the tree may contain redundant splits.

As in [Wallace and Patrick, 1993], there is no cost for stating leaf node parameters and the classifications in these leaf nodes are coded using a sequential code.

4.5 An Attribute Selection Agent

A final simple agent will be defined. This agent wants an overview of how probable it is that each attribute influences the classification. Unlike the MML-agent, it does not need a tree estimate nor does it need to be able to make predictions about future data.

For this agent the conclusion takes the form of a vector $w \in [0,1]^M$ where w_j is the probability that attribute w_j appears in some split node of the true tree. The decision forest selection algorithm developed for the predictive agent (see section 4.3) can be used to make this inference. Sample T trees randomly from distribution $\Pr(\Theta|z)$, record the frequency with which each attribute appears in the sampled forest (counting no attribute more than once per tree), then derive the estimate of w from those frequencies.

This sort of estimate can be useful when the goal is not to predict (as with the predictive agent) nor to find the single best complete model (as with the MML-agent), but to direct future experiments. It has value as an explanation, but aimed at a different purpose compared to the MML explanation.

4.6 Three Algorithms Separately

Before developing hybrid agent solutions, the algorithms for the three simple agents will be looked at separately to verify that they perform as expected.

It is expected that the MEKLD forest estimate will perform better (according to log-loss) than the MML estimate at predicting validation sets. It is also expected that MML estimate is unlikely to contain nodes splitting on attributes that do not appear in the "true" tree.

First, these predictions will be tested using artificially generated data sampled from the assumed a priori model. This has the advantage that the true tree can be known and directly compared to.

100 trees were sampled randomly from the a priori distribution $h(\Theta)$. $M = 12$ binary attributes were used and there are $K = 2$ classes. For each tree a training set was generated as well as a validation set with 400 data items. For each tree a MEKLD forest (MEKLD-f) was selected for the predictive agent (section 4.3) and a MML tree

was selected for the MML-agent (section 4.4).

The results are recorded in tables 4.1, 4.2 and 4.3. The average log-loss on the validation set was recorded for each run for both MEKLD forests and MML trees (column label "LL"). For each run, the absolute value of the size difference (measured in number of internal nodes) between the true tree and the MML tree was recorded (column label "SzErr" shows the average of these values). For each run, the number of attributes present in the MML tree but absent in the true tree was recorded (column label "Att+"). For each run, the number of attributes present in the true but absent in the MML tree was recorded (column label "Att-"). The averages of these measurements over all runs are displayed in the tables 4.1, 4.2 and 4.3; for training sets of size $N \in \{20, 40, 60\}$ respectively.

	N	LL	SzErr	Att+	Att-
True	20	0.499			
MML	20	0.594	3.20	0.02	2.72
MEKLD-f	20	0.567			

Table 4.1: Comparison of algorithms on artificial data size $N = 20$.

	N	LL	SzErr	Att+	Att-
True	40	0.459			
MML	40	0.519	2.56	0.00	2.16
MEKLD-f	40	0.500			

Table 4.2: Comparison of algorithms on artificial data size $N = 40$.

	N	LL	SzErr	Att+	Att-
True	60	0.459			
MML	60	0.512	2.43	0.02	1.90
MEKLD-f	60	0.494			

Table 4.3: Comparison of algorithms on artificial data size $N = 60$.

Notice that, as expected, the MEKLD forests do predict better than the MML trees. Notice also that the MML estimator is very conservative, it almost never splits on an attribute not present in the true tree.

It it is harder to find a measure or experiment that directly verifies that the method for inferring parameter w for the attribute selection agent (section 4.5) works. Since the value selected for w is read directly from the MEKLD forest selected for the predictive agent, the predictive performance of that method gives evidence that the sampling method works as intended; which, in turn, supports our method for selecting w.

4.7 A First Hybrid-Solution

A hybrid of the MEKLD and MML solutions will now be created. The first obstacle is that our MEKLD solution produces a forest while the MML solution produces a tree. A conclusion language is needed which both agents can use. For any set $\phi \in \Theta^T$ of T trees, there exists a tree $\theta \in \Theta$ such that $\Pr(y_{N_1}|x_{N+1}, \theta) = \phi(y_{N_1}|x_{N+1})$ (where $\phi(y_{N_1}|x_{N+1})$ is defined according to uniform weighting as in equation 4.1 section 4.3). It should therefore be possible to project any given forest estimate ϕ onto the space of trees Θ in a way that preserves predictive power – at least for a single future data item.

Such a projected tree can be created by growing it one node at a time in a nested algorithm. The attribute and value to split on in each iteration will be the one that minimises the expected Kullback-Leibler divergence from forest ϕ to the candidate tree. This value can be approximated by first sampling a large set of hypothetical future data items x^* (attributes only, not classifications) according to the training set and a prior model described in section 4.2. The KL divergence from forest ϕ to tree candidate θ is then calculated as the average KL divergence over the members of x^*.

Our first hybrid solution will grow a single tree in this way according to the MEKLD objective. As before, artificial data was used to verify that the new hybrid works as intended. It is expected that the hybrid tree will predict as well (or nearly as well due to our approximation method) as the MEKLD forest.

The artificial data experiment used in section 4.6 was performed using the new MEKLD tree estimator (MEKLD-t). The results are displayed in tables 4.4, 4.5 and 4.6.

	N	LL	SzErr	Att+	Att-
True	20	0.499			
MML	20	0.594	3.20	0.02	2.72
MEKLD-f	20	0.567			
MEKLD-t	20	0.572	11.6	4.86	1.02

Table 4.4: Performance of the first hybrid solution, data size $N = 20$.

	N	LL	SzErr	Att+	Att-
True	40	0.459			
MML	40	0.519	2.56	0.00	2.16
MEKLD-f	40	0.500			
MEKLD-t	40	0.505	12.1	5.21	0.67

Table 4.5: Performance of the first hybrid solution, data size $N = 40$.

	N	LL	SzErr	Att+	Att-
True	60	0.459			
MML	60	0.512	2.43	0.02	1.90
MEKLD-f	60	0.494			
MEKLD-t	60	0.499	12.1	4.96	0.49

Table 4.6: Performance of the first hybrid solution, data size $N = 60$.

It can be seen that the MEKLD tree predicts nearly as well as the MEKLD forest and better than the MML tree. This comes at a cost though; the MEKLD tree always splits on as many nodes as it can. A hard limit had to be placed on possible tree depth to stop them from becoming unmanageably big. The MEKLD trees, while good predictively, are difficult to interpret and arguably poor as explanations. They never resemble the true tree in size or structure.

4.8 A Better Hybrid-Solution

The MML estimator is very conservative in terms of tree size. If some attribute is split on in the MML tree it is almost always in the true tree (for the artificial data considered in sections 4.6 and 4.7). Sometimes this cautiousness is not what is desired. Consider the case where the goal is to find what attributes might have influence so that future data collection may be better directed. In that case, it can help to find a tree that is slightly more speculative. The MEKLD tree from the previous section (4.7) is too speculative in terms of what attributes it chooses to split on. Having a hybrid of these two methods could help give more flexibility to the process of selecting explanations.

To achieve this we look to the "mixed objectives" concept described in section 4.1. The objective for the MML estimator is a two-part encoding length of the training set z. Let $G_1(\theta, z)$ denote this objective for observation z and hypothesis θ. It is equal to, $G_1(\theta, z) = G_1(\theta) - \log(\Pr(y|x, \theta))$. Here $G_1(\theta)$ is the coding length of tree θ (the assertion length described in section 4.4). The term $-\log(\Pr(y|x, \theta))$ is called the detail and it represents the cost of encoding the observation after the selected hypothesis θ is known.

The objective for the MEKLD tree estimator is the expected KL divergence from $\Pr(y_{N_1}|x_{N+1}, z)$ to $\Pr(y_{N_1}|x_{N+1}, \theta)$ which will be denoted by $G_2(\theta, z)$. Note that the units of both objectives G_1 and G_2 are in Shannon information. This suggests that they might be traded off against each other. The value $G_2(\theta, z)$ is an expected coding cost for a single future data item; the value $G_1(\theta, z)$ is a coding cost for the N observed data items. We will weigh these objectives against each other using the function,

$$G_3(\theta, z) = G_1(\theta, z)\frac{1-r}{N} + G_2(\theta, z)r \tag{4.2}$$

where r is a compromise parameter in the range $[0, 1]$. Since both the MML tree and the MEKLD tree algorithms work using a nested node growing algorithm, a new algorithm can be made for the hybrid which splits nodes according to objective G_3.

Note that there have been past applications of merging MML/MDL with other criteria. One example is [Konagaya and Kondou, 1993] where MDL was merged with maximum likelihood using a real valued parameter to control. Our motivation are how-

ever different in that we are merging criteria to control explanatory speculation (see the second last paragraph of this section).

The artificial data experiment used in section 4.6 was performed using the new hybrid tree estimator. The results are displayed in tables 4.7, 4.8 and 4.9. Here compromise parameter values $r \in \{0.2, 0.5, 0.8\}$ were used.

	r	N	LL	SzErr	Att+	Att-
True		20	0.499			
MML		20	0.594	3.20	0.02	2.72
MEKLD-f		20	0.567			
MEKLD-t		20	0.572	11.6	4.86	1.02
Hybrid	0.8	20	0.580	3.00	0.14	2.55
Hybrid	0.5	20	0.580	3.11	0.04	2.63
Hybrid	0.2	20	0.584	3.16	0.02	2.68

Table 4.7: Performance of the improved hybrid solution, data size $N = 20$.

	r	N	LL	SzErr	Att+	Att-
True		40	0.459			
MML		40	0.519	2.56	0.00	2.16
MEKLD-f		40	0.500			
MEKLD-t		40	0.505	12.1	5.21	0.67
Hybrid	0.8	40	0.510	2.25	0.06	1.87
Hybrid	0.5	40	0.515	2.43	0.02	2.05
Hybrid	0.2	40	0.520	2.55	0.00	2.15

Table 4.8: Performance of the improved hybrid solution, data size $N = 40$.

It can be seen that the hybrid performed well both predictively, and in correctly estimating the tree size and which attributes were split on. The "SzErr" column records how far off the estimated tree size was from the true tree size on average.

Since MML is conservative in choosing when to split a node, it can be expected to systematically underestimate the tree size. MEKLD will overestimate the tree size. As r increases from 0 to 1, it behaves more like MEKLD and less like MML. One would therefore predict that as r rises from 0, the average tree size estimated will increase.

	r	N	LL	SzErr	Att+	Att-
True		60	0.459			
MML		60	0.512	2.43	0.02	1.90
MEKLD-f		60	0.494			
MEKLD-t		60	0.499	12.1	4.96	0.49
Hybrid	0.8	60	0.501	2.04	0.16	1.60
Hybrid	0.5	60	0.506	2.30	0.04	1.80
Hybrid	0.2	60	0.511	2.43	0.02	1.90

Table 4.9: Performance of the improved hybrid solution, data size $N = 60$.

At first it should increase to where it is similar to the average actual tree size. As it gets closer to 1, it will eventually, on average, overestimate the tree size. It is therefore expected that (moving from $r = 0$ to $r = 1$) "SzErr" will first decrease from the MML value and at some point turn around and increase to the MEKLD value. Looking at the results above, the turn-around point must lie somewhere above $r = 0.5$ but it is not clear if it is higher than $r = 0.8$.

As r increases, the hybrid 'speculates' more – in the sense that it is more likely to split on attributes for which it is not that clear if they are relevant. This is why "Att+" (the number of incorrectly include attributes) increases as r increases. Similarly, as r increases, the hybrid is less likely to fail to include attributes that are in the true tree as MEKLD prefers to include more node splits; this is why "Att-" decreases as r increases.

Given these properties, r can be thought of as a parameter that tells the hybrid tree how much it is allowed to 'speculate' about the tree structure. Higher values of r lead to larger trees and are better when the cost of failing to identify relevant attributes is more important. Lower values of r lead to smaller trees and are better when avoiding incorrectly including irrelevant attributes is more important.

4.9 A Final Hybrid-Solution

To finish the demonstration, the hybrid MEKLD/MML from the previous section will be combined with the attribute selection method described in section 4.5. Remember that there the inference is simply a vector $w = (w_1, w_2, \ldots, w_M)$ with each w_j stating how probable it is that the true tree splits on attribute j at least once.

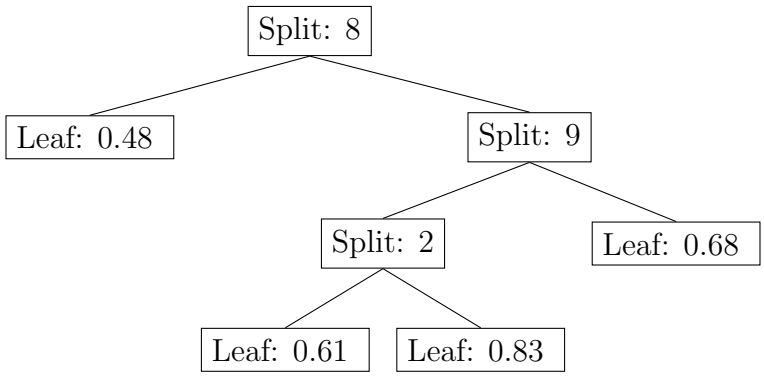
The MEKLD/MML hybrid selects a single tree $\theta \in \Theta$; there is no obvious way to read w off of θ. Attribute j either appears in θ or it doesn't. Reading w directly off this estimate would lead to w being a series of ones and zeros.

The information contained in the tree estimate will be adjusted such that a single tree can be interpreted as a distribution over several trees. First, another compromise parameter $t \in (0, 1]$ will be introduced. Setting $t = 0.3$, for example, will prevent attribute j appearing in the tree estimate if $w_j < t = 0.3$. Within this constraint, the MEKLD/MML tree will be grown as before (described in section 4.8).

The set of trees Θ will be replaced with a set of stochastic trees Θ'. For each node in a stochastic tree, whether it is a leaf or split is defined as a value $a \in [0, 1]$. When $a = 1$, it is a leaf with certainty and leaf parameters must be specified for it. When $a = 0$, it is a split parameter with certainty and it contains the parameters describing what attribute to split on as well as two child nodes (sub-trees). When $0 < a < 1$, the node contains leaf parameters, split parameters and two child nodes. With this, a given node can be interpreted as being a leaf with some probability or being a split with some probability. These trees tend to be easy to interpret because the MEKLD/MML hybrid tree algorithm usually has the more probable nodes at lower depth than the less probable ones.

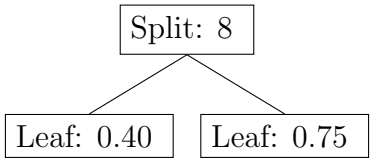
Below is a randomly generated decision tree (figure 4.1). The text "Split: 8" indicates a split on binary attribute $j = 8$. The text "Leaf: 0.61" indicates a leaf node with probability 0.61 for the first class and 0.39 for the second class.

Figure 4.1: True Tree:



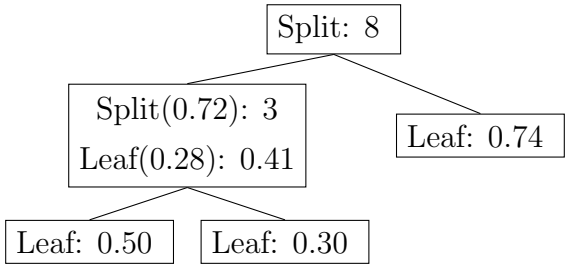
Next is the inferred MML tree using $N = 200$ data items (figure 4.2). The split on attribute 8 is detected but the remainder of the tree is missed.

Figure 4.2: MML Tree:



Our final hybrid was run on the same data using compromise parameters $r = 0.5$ and $t = 0.3$. The result is shown in figure 4.3. Here, one of the nodes is both a split and a leaf; it shows "Split(0.72): 3" to indicate that it is a split on attribute 3 with probability 0.72, and it shows "Leaf(0.28): 0.41" to indicate that it is a leaf with probability 0.28 leading to a 0.41 chance of classification to class one. The hybrid tree detects the split on attribute 8 (as the MML tree did) and it also speculates that a split on attribute 3 might help further down. This turns out to be incorrect, but it was designed to speculate (see final paragraph of section 4.8).

Figure 4.3: Final Hybrid Tree:



Chapter 5

Conclusion

Constructing an Interest-Relative Account of Induction

We have presented arguments for why a more explicitly interest-relative account of inductive inference is needed.

Current methods for formulating inductive inference problems are not well suited to representing the general relation between interest and induction. Chapter 1 has shown that there are quite a lot of seldom considered concepts that must first be developed before even a rudimentary definition of interest-relative induction can be clearly posited.

We have suggested a small set of assumptions on which an interest-relative account of inductive inference could be built:

A) Prior belief is needed for inductive inference.

B) All observations have discrete outcomes.

C) The value and meaning of a belief is derived only from how it influences the actions of agents in environments.

D) Only practical limitations can be used to defend the use of belief distorting inference methods.

E) With the practical limitations taken into consideration, only the environment in which the conclusion is to be used can be used to defend preference between belief distorting inference methods.

In chapter 2 it was demonstrated that an unambiguous and formal model of this relation can be built on these assumptions. That was accomplished by combining a variation of Bayesian probabilities with decision theory. This model has turned out to be more complicated than is desirable and we suspect that this is inevitable given the goals behind it. It needs to represent the full range of possible agent limitations. It needs to represent how the meanings of beliefs may be related among different limited agents and to their hypothetical behaviour. It needs to represent the full range of agent beliefs and agent environments.

Concerning Explanation

The question, when and why are explanations needed? is usually the domain of philosophy of science. Our approach has provided a perspective derived from the technical considerations that arise when placing inference in a realistic context. Explanation is a class of inference that is concerned with explicitly identifying underlying mechanisms and causes of a phenomenon. In section 3.1 arguments were put forward for why explanation should be considered intrinsically interest-relative. In other words, it makes no sense to talk about explanation outside of any realistic context – without considering both practical limitations and practical investment.

The basic relation between practical limitations (of agents), environments of interest and explanation was revealed. As agent limitations become more severe, the environment of interest must be specified more precisely for good inferences to be possible. If all practical constraints are removed, only the environment of interest's minimal sufficient statistic needs to be known. As practical constraints become more severe, more explanation is needed – more needs to be explicitly inferred about the underlying mechanisms of the phenomena being observed. If all practical constraints are removed, no explicit explanation is needed.

One type of practical constraint that may apply to an inference task is conclusion brevity – where there is a limit to how much information can be communicated when communicating the conclusion of an inference. For example, a scientific article in a journal may have a brevity constraint in the form of a page limit. It was argued that for everyday inferences and communications it may be the case that amplicative and possibly explaining inferences may be preferable because of conclusion brevity constraints but,

for formal/academic contexts this is unlikely to be the case. This means that the need for explanation in scientific publications cannot be justified by appealing to brevity constraints.

By eliminating alternatives, it becomes possible to conclude that in formal/academic contexts, explanations are made primarily to reduce future computation (section 3.1). The desire to keep conclusions brief is secondary. Computational difficulty in making inferences is secondary.

Concerning Minimum Message Length Inference

The interest-relative account of inductive inference predicts that any inference principle aimed at guiding inference method design will perform poorly at explanation if it does not allow interest information to enter the problem specification.

The minimum message length principle defines what the best inference is for a given observation and prior belief about the data source. MML is not explicitly defined as an interest relative method. Section 3.5 showed how interest information can – and does in practice – enter the problem specification under minimum message length inference. This happens firstly through the choice of parametrisation and secondly through the way that coding groups are defined. It is argued that for MML to produce decent explanations, it is necessary that the form of allowed coding groups be restricted beyond what would be ideal if one were simply attempting to compress as much as is possible in a two-part scheme. Essentially, hypotheses which are conceptually dissimilar should be prevented from appearing in the same coding group. Conceptually dissimilar here means that, despite being similar according to the likelihood that they define, the hypotheses are acted on differently due to practical limitations of the agents which are to use the inferred estimate.

Concerning Practical Interest-Guided Inference Method Design

It was argued that if some approach to inference algorithm design is meant for producing explanations then it must allow interest information to enter the problem specification (section 3.5). Overly ideal (interest absolute) assumptions about the goals of explanation will lead to inferences that do not have the properties typically considered essential

to explanation. Inductive inference is often about achieving a compromise between requirements. Many existing approaches to inference method design make this difficult by tacitly forcing overly ideal assumptions about practical limitations and practical investment.

The model of interest-relative induction presented in chapter 2 is too complex to be applied directly to inference algorithm design. In chapter 4 some principles for practical interest specification were introduced under the label *the hybrid-agent method*.

The core idea is that a complex interest might be more easily specified if it were described as a compromise between multiple simpler agents; each having a more pure and easily described interest. One starts by designing separate inference algorithms for each agent. After that, these solutions are merged into a hybrid algorithm. Two principles for achieving such hybrids were demonstrated: *parallel interpretations* and *mixed objectives*.

Parallel interpretations is the notion that different agents may interpret a given conclusion in different ways; at different or incomparable levels of understanding. One must look for explanations that are good under multiple interpretations.

Mixed objectives is the notion that a single conclusion to be used by different agents can usually only be achieved if there is some method for weighing their objectives against each other.

Application of the hybrid-agent method for inference algorithm design was demonstrated in chapter 4 using decision tree problems as an example. This has also lead to the creation of some new and decision tree inference algorithms.

Future Work

The method for defining belief and shared meaning for limited agents (from section 1.22) might be improved by more explicitly representing hypothetical communication between agents. As it is currently, there is only one "code" for each possible belief, which is unrealistic.

While an explanation for how interest information enters the problem specifications for MML inference was presented, it should be possible to demonstrate how this can be used to help design new MML methods.

The hybrid-agent method for inference algorithm design presented in chapter 4 shows promise and we hope to apply it to more problems.

Agent practical limitations were represented in our model as hard constraints. It should be possible to reformulate it such that there is, instead, also a quantitative penalty for agent deliberation. Such definitions are not difficult to make but we have not yet found a way to do it such that the implications can be easily investigated. Changing to quantitative acting constraints might allow for the worst case expected utility measure used in our model (section 2.6) to be replaced with something more desirable – not determined by the worst case alone – since they may then have preferences (over beliefs) due to some being more easily acted on than others.

Bibliography

A. Bales, D. Cohen, and T. Handfield. Decision theory for agents with incomplete preferences. *Australasian Journal of Philosophy*, 92(3):453–470, 2014.

J. Baron. *Rationality and intelligence*. Cambridge University Press, 2005.

J. O. Berger, J. M. Bernardo, and D. Sun. The formal definition of reference priors. *The Annals of Statistics*, 37(2):905–938, 2009.

D. M. Boulton and C. S. Wallace. An information measure for single link classification. *The Computer Journal*, 18(3):236–238, 1975.

D. V. Budescu and T. S. Wallsten. Processing linguistic probabilities: General principles and empirical evidence. In J. Busemeyer, R. Hastie, and D. L. Medin, editors, *The psychology of learning and motivation: Decision Making from the Perspective of Cognitive Psychology*, volume 32, pages 275–318. Academic Press, New York, 1995.

G. J. Chaitin. On the length of programs for computing finite binary sequences. *Journal of the ACM (JACM)*, 13(4):547–569, 1966.

J. W. Comley and D. L. Dowe. General Bayesian networks and asymmetric languages. In *Proc. Hawaii International Conference on Statistics and Related Fields*, pages 5–8, 2003.

J. W. Comley and D. L. Dowe. Minimum message length and generalized Bayesian nets with asymmetric languages. In P. D. Grünwald, I. J. Myung, and M. A. Pitt, editors, *Advances in Minimum Description Length: Theory and Applications*, chapter 11, pages 265–294. MIT Press, 2005.

D. L. Dowe. Foreword re C.S. Wallace. *The Computer Journal*, 51(5):523–560, 2008.

D. L. Dowe. MML, hybrid Bayesian network graphical models, statistical consistency, invariance and uniqueness. In P. S. Forster and M. R. Bandyopadhyay, editors, *Philosophy of Statistics*, volume 7 of *Handbook of the Philosophy of Science*, pages 901–982. Elsevier, 2011.

D. L. Dowe. Introduction to Ray Solomonoff 85th memorial conference. In *Algorithmic Probability and Friends. Bayesian Prediction and Artificial Intelligence*, volume 7070 of *LNCS*, pages 1–36. Springer, 2013.

D. L. Dowe, R. A. Baxter, J. J. Oliver, and C. S. Wallace. Point estimation using the Kullback-Leibler loss function and MML. In *2nd Pacific-Asia Conference on Knowledge Discovery and Data Mining*, volume 1394 of *LNAI*, pages 87–95, Melbourne, Australia, April 1998. Springer.

D. L. Dowe, J. Hernández-Orallo, and P. K. Das. Compression and intelligence: social environments and communication. In *Artificial General Intelligence*, volume 3830 of *LNAI*, pages 204–211. Springer, 2011.

J. G. Dowty. SMML estimators for linear regression and tessellations of hyperbolic space. *arXiv preprint arXiv:1403.2201*, 2014.

J. G. Dowty. SMML estimators for 1-dimensional continuous data. *The Computer Journal*, 58(1):126–133, 2015.

D. Fink. A compendium of conjugate priors. Technical report, Environmental Statistical group, Department of Biology, Montana State University, USA, 1997.

L. J. Fitzgibbon, D. L. Dowe, and L. Allison. Univariate polynomial inference by Monte Carlo message length approximation. In *Proceedings of the 19th International Conference on Machine Learning (ICML 2002)*, volume 2417, pages 147–154, Sydney, Australia, 2002. Morgan Kaufmann.

R. N. Giere. Using models to represent reality. In L. Magnani, N. J. Nersessian, and P. Thagard, editors, *Model-based reasoning in scientific discovery*, pages 41–57. Springer US, Boston, MA, 1999.

R. N. Giere. How models are used to represent reality. *Philosophy of science*, 71(5): 742–752, 2004.

P. D. Grünwald. *The Minimum Description Length Principle*. The MIT Press, January 2007.

J. Hernández-Orallo and D. L. Dowe. Measuring universal intelligence: Towards an anytime intelligence test. *Artificial Intelligence*, 174(18):1508–1539, 2010.

J. Hernandez-Orallo and N. Minaya-Collado. A formal definition of intelligence based on an intensional variant of algorithmic complexity. In *Proceedings of International Symposium of Engineering of Intelligent Systems (EIS98)*, pages 146–163, 1998.

R. Holton. Partial belief, partial intention. *Mind*, 117(465):27–58, 2008.

R. Holton. Intention as a model for belief. In M. Vargas and G. Yaffe, editors, *Rational and Social Agency: Essays on the Philosophy of Michael Bratman*. Oxford University Press, 2013.

M. Hutter. *Universal artificial intelligence: Sequential decisions based on algorithmic probability*. Springer, Berlin, 2005.

T. Jebara. *Discriminative, Generative and Imitative Learning*. PhD thesis, Massachusetts Institute of Technology, 2001.

D. Kahneman and A. Tversky. *Choices, values, and frames*. Cambridge University Press, 2000.

J. G. Kemeny. Fair bets and inductive probabilities. *The Journal of Symbolic Logic*, 20 (03):263–273, 1955.

A. N. Kolmogorov. *Foundations of the Theory of Probability*. Chelsea Publishing Co., 1950.

A. N. Kolmogorov. Three approaches to the quantitative definition of information. *Problems of information transmission*, 1(1):1–7, 1965.

A. Konagaya and H. Kondou. Stochastic motif extraction using a genetic algorithm with the mdl principle. In *Proceedings of the Twenty-sixth Hawaii International Conference on System Sciences*, volume 1, pages 746–755, Jan 1993.

T. S. Kuhn. *The Structure of Scientific Revolutions*. University of Chicago press, 1962.

S. Legg. *Machine super intelligence*. PhD thesis, Department of Informatics, University of Lugano, 2008.

D. J. C. MacKay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

P. M. Murphy and M. J. Pazzani. Exploring the decision forest: An empirical investigation of Occam's razor in decision tree induction. *Journal of Artificial Intelligence Research*, 1:257–275, 1994.

S. L. Needham and D. L. Dowe. Message length as an effective Ockham's razor in decision tree induction. In *Proc. 8th International Workshop on Artificial Intelligence and Statistics (AI+ STATS 2001)*, pages 253–260, 2001.

J. Pearl. *Causality*. Cambridge university press, 2009.

K. Popper. *The logic of scientific discovery*. Routledge, 2014.

F. P. Ramsey. Truth and probability (1926). *The foundations of mathematics and other logical essays*, pages 156–198, 1931.

J. Rissanen. Modeling by shortest data description. *Automatica*, 14(5):465–471, 1978.

G. Shmueli. To explain or to predict? *Statistical Science*, 25(3):289–310, 2010.

R. J. Solomonoff. A formal theory of inductive inference. *Information and Control*, 7: 1–22,224–54, 1964.

R. J. Solomonoff. Does algorithmic probability solve the problem of induction? In D.L. Dowe, K.B. Korb, and J.J. Oliver, editors, *Proceedings of the Information, Statistics and Induction in Science (ISIS) Conference*, pages 7–8, Melbourne, Australia, August 1996. World Scientific.

J. Stanley. *Knowledge and practical interests*. Oxford University Press, 2005.

P. J. Tan and D. L. Dowe. MML inference of oblique decision trees. In *AI 2004: Advances in Artificial Intelligence: 17th Australian Joint Conference on Artificial Intelligence*, volume 3339 of *LNAI*, pages 1082–1088, Cairns, Australia, December 2004. Springer.

G. Visser and D. L. Dowe. Minimum message length clustering of spatially-correlated data with varying inter-class penalties. In *Proc. 6th IEEE International Conference on Computer and Information Science (ICIS 2007)*, pages 17–22, Melbourne, Australia, July 2007.

G. Visser, D. L. Dowe, and I. Svalbe. Information-theoretic image reconstruction and segmentation from noisy projections. In *Australasian Conference on Artificial Intelligence*, pages 170–179, 2009a.

G. Visser, D. L. Dowe, and P. Uotila. Enhancing MML clustering using context data with climate applications. In *Australasian Conference on Artificial Intelligence*, pages 350–359, 2009b.

G. Visser, P. Dale, D. L. Dowe, E. Ndoen, M. Dale, and N. Sipe. A novel approach for modeling malaria incidence using complex categorical household data: The minimum message length (MML) method applied to Indonesian data. *Computational Ecology and Software*, 2(3):140–159, September 2012.

C. S. Wallace. On the selection of the order of a polynomial model. Technical report, Royal Holloway College, England, U.K., 1997. see www.csse.monash.edu.au/dld/CSWallacePublications.

C. S. Wallace. Intrinsic classification of spatially correlated data. *Computer Journal*, 41 (8):602–611, 1998.

C. S. Wallace. *Statistical and inductive inference by minimum message length*. Springer, 2005.

C. S. Wallace and D. M. Boulton. An information measure for classification. *The Computer Journal*, 11(2):185–194, 1968.

C. S. Wallace and D. L. Dowe. Intrinsic classification by MML-the Snob program. In *Proceedings of the 7th Australian Joint Conference on Artificial Intelligence*, pages 37–44. World Scientific, November 1994.

C. S. Wallace and D. L. Dowe. Minimum message length and Kolmogorov complexity. *Computer Journal*, 42(4):270–283, 1999.

C. S. Wallace and D. L. Dowe. MML clustering of multi-state, Poisson, von Mises circular and Gaussian distributions. *Statistics and Computing*, 10(1):73–83, 2000.

C. S. Wallace and P. R. Freeman. Estimation and inference by compact coding. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 240–265, 1987.

C. S. Wallace and J. D. Patrick. Coding decision trees. *Machine Learning*, 11(1):7–22, 1993.

P. Walley. Towards a unified theory of imprecise probability. *International Journal of Approximate Reasoning*, 24(2):125–148, 2000.

G. I. Webb. Further experimental evidence against the utility of Occam's razor. *Journal of Artificial Intelligence Research*, 4:397–417, 1996.

D. H. Wolpert and W. G. Macready. No free lunch theorems for optimization. *Evolutionary Computation, IEEE Transactions on*, 1(1):67–82, Apr 1997.

Appendix A

Some Details for Chapter 2

A.1 Incomplete Belief Conditionals

Here is an outline of the proof for equation 2.4 from section 2.4.

$$
\begin{aligned}
(\phi \oplus e) \oplus d &= \{\, p \in P \mid \exists q \in (\phi \oplus e),\ p(\cdot) = q(\cdot|d) \,\} && \text{(A.1)}\\
&= \{\, p \in P \mid \exists r \in \phi,\ \exists q \in P,\ q(\cdot) = r(\cdot|e),\ p(\cdot) = q(\cdot|d) \,\} && \text{(A.2)}\\
&= \{\, p \in P \mid \exists r \in \phi,\ p(\cdot) = r(\cdot|e \cap d) \,\} && \text{(A.3)}\\
&= \phi \oplus (e \cap d) && \text{(A.4)}\\
&= (\phi \oplus d) \oplus e && \text{(A.5)}
\end{aligned}
$$

Remember that $\phi \oplus e$ is meant to be an analogy to conditional probability but for incomplete beliefs. For complete belief p one writes $p(\cdot|e)$ while for incomplete belief ϕ one writes $\phi \oplus e$. $\phi \subseteq P$ is a set of probability functions while $e \in E$ is some event. P is the set of all probability functions that can be defined for event set E. The operation $\oplus$ was defined in section 2.4 as, $\phi \oplus e \ = \ \{\, p \in P \mid \exists q \in \phi,\ p(\cdot) = q(\cdot|e) \,\}$. Note that $\phi \in \Phi$ and $\phi \oplus e \in \Phi$.

A.2 Generated Incomplete Belief Set

Here is an outline of the proof for equation 2.5 from section 2.4.

$$(\Gamma \otimes D) \otimes G = \{\, \phi \in \Phi \mid \exists g \in G,\ \exists \gamma \in (\Gamma \otimes D),\ \phi = \gamma \oplus g \neq \emptyset \,\} \tag{A.6}$$

$$= \{\, \phi \in \Phi \mid \exists g \in G,\ \exists d \in D,\ \exists \gamma' \in \Gamma,\ \phi = (\gamma' \oplus d) \oplus g \neq \emptyset \,\} \tag{A.7}$$

$$= \{\, \phi \in \Phi \mid \exists\, (d \cap g) \in D \cdot G,\ \exists \gamma' \in \Gamma,\ \phi = \gamma' \oplus (d \cap g) \neq \emptyset \,\} \tag{A.8}$$

$$= \Gamma \otimes (D \cdot G) \tag{A.9}$$

$$= (\Gamma \otimes G) \otimes D \tag{A.10}$$

Remember that $\Gamma \otimes D$ is method for generating agent languages. Γ is a set of incomplete beliefs where each $\gamma \in \Gamma$ is a set of probability functions, $\gamma \in \Phi = \mathcal{P}(P)$. D is a set of events $D \subseteq E$. The motivation for $\Gamma \otimes D$ is that it represents an agent language where Γ is the set of unobservable models that the agent's mind may represent (agent hypotheses, see section 2.14) while D is the set of observable events that the agent's mind may represent. If the agent believes unobservable model $\gamma \in \Gamma$ and observable event $d \in D$ then it believes $\gamma \oplus e$ which is a member of $\Gamma \otimes D$ – unless e contradicts γ. The operation was defined in section 2.4 as, $\Gamma \otimes D = \{\, \phi \in \Phi \mid \exists d \in D,\ \exists \gamma \in \Gamma,\ \phi = \gamma \oplus d \neq \emptyset \,\}$. Note that $\Gamma \subseteq \Phi$ and $\Gamma \otimes D \subseteq \Phi$.

A.3 Acting Constraints and Belief Distortion

Below are outlines of the proofs for the lemmas from section 2.11.

For **lemma 1,** define an agent/environment holding the following properties. Let $b_1 \in B_i$ be the only optimal behaviour for $\phi = \{\sigma\} \in \Phi$. Let $\gamma \in \Phi$ be some belief such that $\{\sigma\} \neq \gamma$ and for which b_1 is also the only optimal behaviour. Restrict U_i such that $b_1 \notin U_i(\phi)$ while $b_1 \in U_i(\gamma)$. Now, (σ, γ) is a better inference than $(\sigma, \{\sigma\})$ even though $\phi \in \Phi_i$.

For **lemma 2,** assume that $\{\sigma\} \in \Phi_i$ and that U_i is separable. Assume that for $\phi \in \Phi_i$, $s_i(\sigma, \phi) > s_i(\sigma, \{\sigma\})$. This can only happen when there is some $b \in U_i(\phi)$ while $b \notin U_i(\{\sigma\})$ where the expected utility of b under σ is greater than for any $b' \in U_i(\{\sigma\})$. Since $\{\sigma\}$ totally orders all behaviours in B_i, there exists some $B' = U_i(\{\sigma\}) \cap b$ for which $U_i(\{\sigma\}) \subset B'$ while $\forall \delta > 0, \{\sigma\} \in \kappa_i(B', \delta)$. This contradicts definition 15 (of separability) so the assumption that U_i is separable is contradicted.

For **lemmas 3 and 4:** For all $p \in P$, $\{p\}$ always totally orders all behaviours B_i. That leads to the following, $\forall B' \subseteq B_i,\ B' \neq \emptyset \ \Rightarrow\ \forall p \in P,\ \{p\} \in \kappa_i(B', 0)$. As a result, $\{p\}$ is in Φ_i for all restricted and relaxed acting constraints for all $p \in P$ – following definitions 16 and 17. Since all restricted and relaxed acting constraints are separable, the result for lemmas 3 and 4 then follows from lemma 2.

A.4 Order of Environments

Below are details for the lemmas from sections 2.12 and 2.13.

For **lemma 7,** we will construct the desired environment $\mathcal{E}_i$ with minimal sufficient statistic Z. Let $g > 0$ be some real valued number while $p \in P$. Define condition $c(p, g, Z) = 1$ when $\forall z \in Z,\ p(z) \geq g$, otherwise $c(p, g, Z) = 0$. Define condition $c'(p, g, Z) = 1$ when $\forall z \in Z,\ p(z) > g$, otherwise $c'(p, g, Z) = 0$. Construct $\mathcal{E}_i$ by defining for each $p \in P$ for which $c(p, g, Z) = 1$ a behaviour b_p where $\forall \omega \in \Omega$, $u_i(\omega, b_p) = \log(p(Z(\omega)))$. Here $Z(\omega)$ is that member $x \in Z$ for which $\omega \in x$. These are the only behaviours defined for $\mathcal{E}_i$. The condition $c(p, g, Z) = 1$ is required because achievable utility must be bounded for worst case expected utility to be defined. Let $\sigma \in P$ be some premise for which $c'(p, g, Z) = 1$. It will be the case that $\{\sigma\} = S(\sigma, 0)$. This is because for any $p \in P$ near σ (staying within the condition $c(p, g, Z) = 1$) there will be a behaviour b_p for which b_p has greater expected utility than b_σ under p. That follows from Shannon's source coding theorem. The environment $\mathcal{E}_i$ can therefore distinguish between all premises in the set $\{\ \sigma \in P \mid c'(p, g, Z)\ \}$. Note that B_i as described above is uncountable, if it were not, $S(\sigma, \nu)$ would be uncountable for all $\nu > 0$; thus, there would always be some pair of premises which $\mathcal{E}_i$ cannot distinguish between for a given value of $\nu > 0$.

For **lemma 8,** define $\mathcal{E}_k$ such that $B_k = B_i \times B_j$ while $u_k(\omega, (b_1, b_2)) = u_i(\omega, b_1) + u_i(\omega, b_2)$ for all $\omega \in \Omega$.

A.5 Amplicative and Destructive Inference

Below are the details for **lemma 10** from section 2.16:

$$\sigma \in \varphi(\phi) \oplus \epsilon(\{\sigma\}) \tag{A.11}$$

$$\Leftrightarrow\ \sigma\ \in\ \big(\ \{\ \lambda \in P \mid \lambda(\cdot|\epsilon(\phi)) \in \phi\ \}\ \oplus\ \epsilon(\{\sigma\})\ \big) \tag{A.12}$$

$$\Leftrightarrow\ \sigma\ \in\ \{\ q \in P \mid \exists \lambda \in P,\ \lambda(\cdot|\epsilon(\phi)) \in \phi,\ q(\cdot) = \lambda(\cdot|\epsilon(\{\sigma\}))\ \} \tag{A.13}$$

$$\Leftrightarrow\ \exists \lambda \in P,\ \big(\ \lambda(\cdot|\epsilon(\phi)) \in \phi\ \big)\ \wedge\ \big(\ \sigma(\cdot) = \lambda(\cdot|\epsilon(\{\sigma\}))\ \big) \tag{A.14}$$

$$\Leftrightarrow\ \exists \lambda \in P,\ \big(\ \lambda(\cdot|\epsilon(\phi)) \in \phi\ \big)\ \wedge\ \big(\ \{\sigma\} = \{\lambda\} \oplus \epsilon(\{\sigma\})\ \big) \tag{A.15}$$

From $\lambda(\cdot|\epsilon(\phi)) \in \phi$ it follows that $\lambda \in \varphi(\phi)$. Letting $\gamma = \varphi(\phi)$ gives $\lambda \in \gamma$ while $\phi = \gamma \oplus \epsilon(\phi)$.

Remember that the purpose of lemma 10 is to show that iff condition A.11 holds – for premise σ and conclusion ϕ – then there will exist some $\lambda \in P$ such that it is possible that λ is the data source hypothesis (see section 2.8) and a member of the acting agent hypothesis γ (see section 2.14).

A.6 Communication Constraints and Amplicativity

For **lemma 11:** Let k be an ι optimal inference algorithm for such a problem, $k \in W_i(\iota)$. Let $g(\sigma)$ denote the set $\{\ p \in \Lambda \mid k(\sigma) = k(p)\ \}$. If $g(\sigma)$ has only one member then using $\{\sigma\}$ as conclusion when σ is the premise is at least as good a conclusion as $k(\sigma)$. This follows since the acting constraint is separable and $\{\sigma\}$ is in Φ_i by assumption. We can replace k with some function k' which is identical except that $k'(\sigma) = \{\sigma\}$ and so avoid distortion for this premise. Assume instead that $g(\sigma)$ has many members. Let $Z \subseteq X$ be the event for which $\lambda(\cdot|z) \in g(\sigma) \Leftrightarrow z \in Z$. When the premise is in $g(\sigma)$, the conclusion $\lambda(\cdot|Z)$ is as good as $k(\sigma)$ since the acting constraint is separable and $\{\lambda(\cdot|Z)\}$ is in Φ_i by assumption. We can replace k with some function k' which is identical except that $\forall p \in g(\sigma),\ k'(p) = \{\lambda(\cdot|Z)\}$. This is O-Dest but not amplicative or U-Dest.

For **lemma 12,** the method is the same as for lemma 11 except that instead of using $\lambda(\cdot|Z)$ as the conclusion we use $\gamma \oplus Z$. Since $\lambda \in \gamma$ this might be U-Dest but not U-Amp or O-Amp.

Appendix B

Some Details for Chapter 3

B.1 Log-Loss Environments

This section gives the reasoning behind the statement: A log-loss environment is what is approached when approximating the "most invested" environment.

Let $\mathcal{E}_{LL}$ denote the log-loss environment. Let X be the observed variable while Y is the sufficient statistic of $\mathcal{E}_{LL}$ (see definition 11). Let λ be the data source hypothesis – so, given observation $x \in X$ the prior belief implies distribution $\lambda(Y|x)$.

Environment $\mathcal{E}_{LL}$ will ask the agent to specify a probability distribution p over variable Y and will penalise the agent by utility $-\log(p(\hat{y}))$ where $\hat{y}$ is the true value of y. In other words, the action set of $\mathcal{E}_{LL}$ corresponds the set of all probability functions that can be defined over random variable Y.

If there are no practical constraints, any optimal inference method will always produce a conclusion ϕ for which $\phi(Y) = \lambda(Y|x)$ where x was observed. This follows from Gibb's inequality.

Let $\mathcal{E}_k$ be some other environment which also has Y as its sufficient statistic. If no practical constraints apply then any inference method for which $\phi(Y) = \lambda(Y|x)$ holds, for all x, will be in the set optimal inference methods (lemma 5).

It follows that when no practical constraints apply, $\mathcal{E}_k$ can be safely replaced with $\mathcal{E}_{LL}$, thus, $\mathcal{E}_{LL}$ is more invested (see section 2.12). Since this property holds for any $\mathcal{E}_k$ with

sufficient statistic Y, the log-loss environment can be described as the "most-invested" environment.

There are two problems here: First, this property breaks down when practical constraints do apply. Second, $\mathcal{E}_{LL}$ as described here is not strictly a valid environment as defined by our model. The problem is that $\mathcal{E}_{LL}$ is not guaranteed to bound utility above and below; expected utility and worst-case utility might not converge for some inference methods.

One might approximate $\mathcal{E}_{LL}$ by something that has utility bounds past some limit and then imagine moving that limit to infinity. For this reason, it is claimed that: A log-loss environment is what is approached when approximating the "most invested" environment.

B.2 Conjugate Priors

This section describes why using conjugate priors allows inference methods that will not be amplicative. For a full definition of conjugate priors see [Fink, 1997]. Let $X = (X_1, X_2, \dots)$ be a series of random variables. Let Θ be some hypothesis space such that the prior belief about X can be written as,

$$\lambda(X) = \int_{\Theta} f(X|\theta)h(\theta)\, d\theta \tag{B.1}$$

where f is a likelihood function. If Θ, f and h together form a conjugate prior, it will be the case that for all j and all $(x_1, x_2, \dots x_j)$ there exists some $\theta \in \Theta$ such that,

$$f(X_{j+1}, X_{j+2}, \dots \,|\, \theta) = \lambda(X_{j+1}, X_{j+2}, \dots \,|\, x_1, x_2, \dots x_j)\ . \tag{B.2}$$

Since $\lambda(X_{j+1}, X_{j+2}, \dots \,|\, x_1, x_2, \dots x_j)$ is by definition a non-amplicative conclusion when $(x_1, x_2, \dots x_j)$ was observed, it follows that there will always be a member of Θ which corresponds to a non-amplicative inference.

Appendix C

Some Details for Chapter 4

C.1 Decision Tree Sampling

The Metropolis-Hastings algorithm is used for sampling decision trees from the posterior distribution $P(\Theta|z)$; Θ is the set of all trees while z is the training dataset. An initial tree is selected by sampling randomly from the prior distribution over all trees $h(\Theta)$. A candidate function $Q(\theta_{t+1}|\theta_t)$ is used to generate a new tree candidate θ_{t+1} from the current tree θ_t. This is a probability distribution so the next candidate θ_{t+1} is generated randomly depending on the current candidate θ_t. An acceptance ratio $r = f(z|\theta_{t+1})/f(z|\theta_t)$ is calculated where z is the training dataset and f is the likelihood function. If $r \geq 1$ then the candidate is accepted, if not, the transition is accepted with probability r. To prevent getting stuck in local optima, we replace r with $r^{\frac{1}{T}}$ where T is a "temperature" which starts at some value $T > 1$ and is slowly lowered to $T = 1$ during the sampling iteration.

The Metropolis-Hastings algorithm requires that candidate function $Q(\theta_{t+1}|\theta_t)$ be symmetric, i.e., $Q(\theta_1|\theta_2) = Q(\theta_2|\theta_1)$. A few tricks are needed to make decision tree mutations that have this property. We use the following mutations to compose Q:

A: select a random node, if it is a leaf then change it to a split, if it is a split then change it to a leaf

B: select a random node, change the attribute that it splits on to a different attribute with all being equally probable

C: select two random nodes that lie on the same path, swap their split parameters

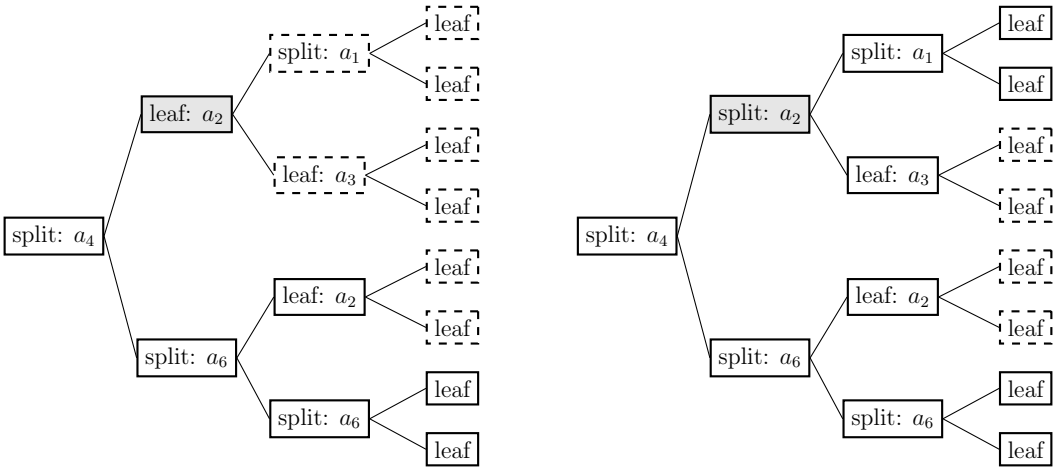
D: select a random node, change the value that the split conditions on to a different value with all values being equally probable

The candidate function implementation randomly selects one of these mutation types in each iteration. Each of these mutations is symmetric with itself, leading to symmetry for Q as a whole. Each mutation is described more carefully in the remaining pages of this section (starting in the next page). These mutations do not involve leaf node parameters – those are derived directly from the training set.

Mutation A**:** *select a random node, if it is a leaf then change it to a split, if it is a split then change it to a leaf.*

Figure C.1 shows a tree before (left) and after (right) application of mutation A to the highlighted node. Mutation A is symmetric with itself: applying it to the same node of the "after" tree returns it to the "before" tree. For each node, there is a label such as "*split:* a_3" which means that the node splits on attribute 3. Note that for a leaf node the a_3 will still be there but it does not affect the resulting likelihood function – we must still keep of that parameter for the mutation to be symmetric. The final level nodes (rightmost) are always leafs and do not need split parameters. Tree depth was capped at maximum 5 levels for the experiments in chapter 4. Unreachable nodes are shown with dashed lines surrounding them. Unreachable nodes do not affect the likelihood function but must be retained in program memory for the mutations to be symmetric. The node to which mutation A is to be applied is selected randomly with each node being equally probable – even unreachable nodes. Notice that on the bottommost path (going $a_4 \to a_6 \to a_6 \to leaf$) the second split on a_6 is redundant. Redundant splits are normally not allowed for decision trees, but for our sampling algorithm to work quickly that requirement was dropped.

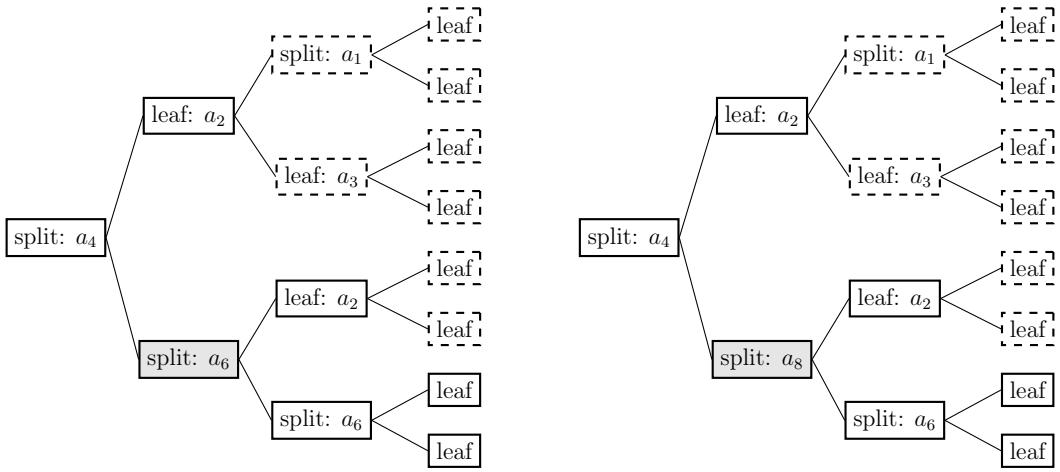
Figure C.1: Mutation A



Mutation B**:** *select a random node, change the attribute that it splits on to a different attribute with all being equally probable.*

Figure C.2 shows a tree before (left) and after (right) application of mutation B to the highlighted node. First, a node is selected selected randomly with each node being equally probable – even leafs and unreachable nodes (shown with dashed lines). Next, an attribute is selected randomly – in this case a_8 – and the split parameter is changed to split on that attribute. If the selected node is a leaf or unreachable node then the likelihood function is unaffected by the mutation. Since the attribute to switch to is selected randomly with uniform chance, the probability of mutation B turning the "before" (left) tree into the "after" (right) tree is the same as the probability of mutation B turning the "after" tree into the "before" tree.

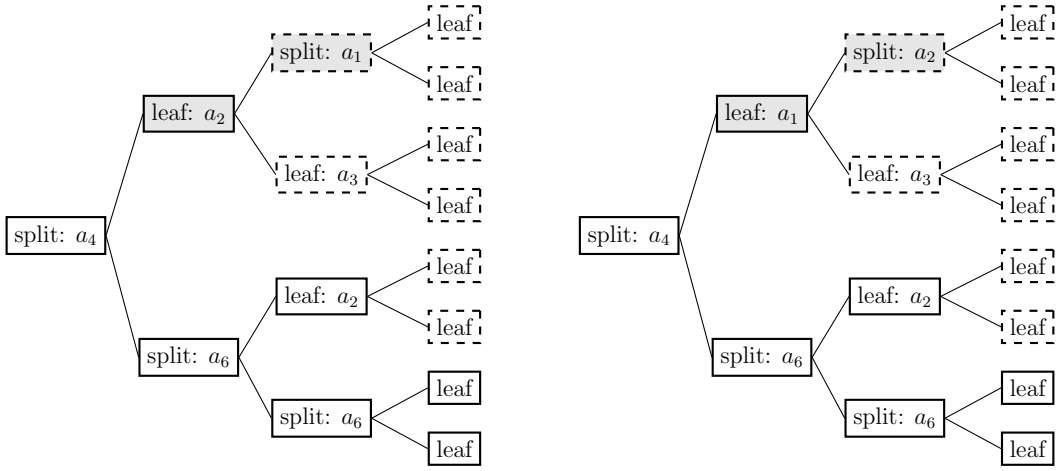
Figure C.2: Mutation B



Mutation C**:** *select two random nodes that lie on the same path, swap their split parameters.*

Figure C.3 shows a tree before (left) and after (right) application of mutation C to the highlighted nodes. First, two nodes lying on the same path (one being the ancestor of the other) are selected randomly. This selection treats leaf nodes, split nodes and unreachable nodes (shown with dashed lines) the same. The mutation is its own inverse when the two nodes selected are the same. Which two nodes are selected does not depend on the state of the tree. Mutation C was included to help reduce the chance of getting stuck in local optima and is not strictly necessary – given mutations A, B and D – for all possible tree states to be reachable.

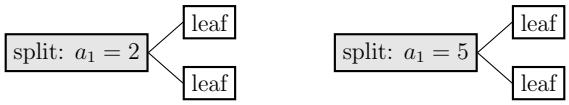
Figure C.3: Mutation C



Mutation D**:** *select a random node, change the value that the split conditions on to a different value with all values being equally probable.*

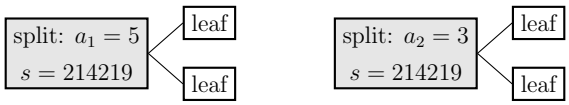
So far, trees have been depicted for which all attributes have only two possible values. Imagine now that attribute a_1 has four possible values $a_1 \in \{1, 2, 3, 5\}$ while attribute a_2 has three possible values $a_2 \in \{1, 2, 3\}$. Figure C.4 shows application of mutation D using value 5. The split condition on the left tree is met if $a_1 = 2$ – e.i. the split leads to the bottom leaf when $a_1 = 2$ and to the top leaf when $a_1 \in \{1, 3, 4, 5\}$. For the right tree, it splits on the condition $a_1 = 5$.

Figure C.4: Mutation D



One problem remains, what happens when the mutations B or C are applied changing a_1 to a_2? Condition $a_1 = 5$ cannot apply to attribute $a_2 \in \{1, 2, 3\}$ so there can be no condition $a_2 = 5$. Mutations B or C must both be symmetric so simply changing $a_1 = 5$ to $a_2 = 3$ will not do as the reverse operation would then lead to $a_1 = 3$. Our solution is to have each node hold a parameter $s \in \{0, 2, \ldots, 2^{64} - 1\}$. This range is determined by the range of a 64 bit integer. Instead of mutation D selecting an value from the set $\{1, 2, 3, 4, 5\}$ for a_1, it selects a random integer s from the range $\{0, 2, \ldots, 2^{64} - 1\}$. The split condition is then $a_1 = 1 + (s \pmod 5)$. If mutation B or C is applied, changing a_1 to a_2, the condition changes to $a_2 = 1 + (s \pmod 3)$ while s is unchanged. Here (mod 5) is used for a_1 because it has five possible values while (mod 3) is used for a_2 because it has three possible values. Figure C.5 shows the result of applying mutation B, changing a_1 to a_2. In this example, mutation B is symmetric because $1 + (21419 \pmod 5) = 5$ while $1 + (21419 \pmod 3) = 3$.

Figure C.5: Mutation B (Revisited)



When the program is given a scalar attribute it makes it discrete by diving it into percentiles (no more than 8) based on the training set distribution. For scalar attributes, or discrete attributes which are marked as ordinal, the "=" condition is replaced by ">"; so, condition $a_1 = 2$ is replaced by $a_1 > 2$ while the boundary cases $a_1 > 1$ and $a_1 > 5$ are removed. Similarly, for any binary valued attribute a_k, the condition $a_k = 1$ is removed as it is equivalent to $a_k = 2$ (with child nodes swapped).