Learning, Bayesian Probability, Graphical Models, and Abduction [1]

David Poole
poole@cs.ubc.ca

Department of Computer Science
University of British Columbia
2366 Main Mall
Vancouver, B.C., Canada V6T 1Z4
http://www.cs.ubc.ca/spider/poole

Abstract

In this chapter I review Bayesian statistics as used for induction and relate it to logic-based abduction. Much reasoning under uncertainty, including induction, is based on Bayes' rule. Bayes' rule is interesting precisely because it provides a mechanism for abduction. I review work of Buntine that argues that much of the work on Bayesian learning can be best viewed in terms of graphical models such as Bayesian networks, and review previous work of Poole that relates Bayesian networks to logic-based abduction. This lets us see how much of the work on induction can be viewed in terms of logic-based abduction. I then explore what this means for extending logic-based abduction to richer representations, such as learning decision trees with probabilities at the leaves. Much of this paper is tutorial in nature; both the probabilistic and logic-based notions of abduction and induction are introduced and motivated.

1 Introduction

This paper explores the relationship between learning (induction) and abduction. I take what can be called the Bayesian view, where all uncertainty is reflected in probabilities. In this paper I argue that, not only can abduction be used for induction, but that most current learning techniques (from statistical learning to neural networks to decision trees to inductive logic programming to unsupervised learning) can be best viewed in terms of abduction.

1.1 Causal and Evidential Modelling and Reasoning

In order to understand abduction and its role in reasoning, it is important to understand ways to model, as well as ways to reason. In this section we consider reasoning strategies independently of learning, and return to learning in Section 1.2.

Many reasoning problems can be best understood as evidential reasoning tasks.

Definition 1.1 An **evidential reasoning task** is where some parts of a system are observed and you want to make inferences about other (hidden) parts.

Example 1.2 The problem of **diagnosis** is an evidential reasoning task. Given observations about the symptoms of a patient or artifact, we want to determine what is going on inside the system to produce those symptoms.

Example 1.3 The problem of **perception** (including **vision**) is an evidential reasoning task. In the world the scene produces the image, but the problem of vision is, given an image, determine what is in the scene.

Evidential reasoning tasks are often of the form where there is a cause-effect relationship between the parts. In diagnosis we can think of the disease *causing* the symptoms. In vision we can think of the scene *causing* the image. By *causation*[2], I mean that different diseases can result in different symptoms (but changing the symptoms doesn't affect the disease) and different scenes can result in different images (but manipulating an image doesn't affect the scene).

There are a number of different ways of modelling such a causal domain:

causal modelling where we model the function from causes to effects. For example, we can model how diseases or faults manifest their symptoms. We can model how scenes produce images.

evidential modelling where we model the function from effects to causes. For example we can model the mapping from symptoms to diseases, or from image to scene.

[2]See `http://singapore.cs.ucla.edu/LECTURE/lecture_sec1.htm` for a fascinating lecture by Judea Pearl on causation.

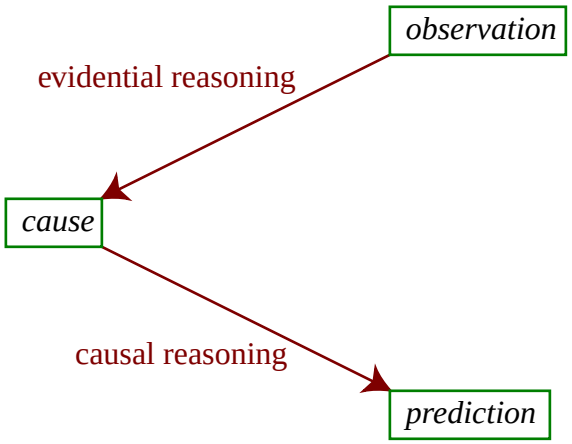

Figure 1: Causal and evidential reasoning

Independently of these two modelling strategies, we can consider two reasoning tasks:

Evidential Reasoning given an observation of the effects, determine the causes. For example, determine the disease from the symptoms, or the scene from the image.

Causal Reasoning given some cause, make a prediction of the effects. For example, predicting symptoms or prognoses from a disease, or predicting an image from a scene. This is often called *simulation*.

In particular, much reasoning consists of evidential reasoning followed by causal reasoning (see Figure 1). For example, a doctor may observe a patient, determine possible diseases, then make predictions of other symptoms or prognoses. This then can feedback to making the doctor look for the presence or absence of these symptoms, forming the cycle of perception [20]. Similarly, a robot can observe its world, determine what is where, and act on its beliefs, leading to further observations.

There are a number of combinations of modelling and reasoning strategies that have been proposed:

- The simplest strategy is to do evidential modelling and only evidential reasoning. Examples of this are neural networks [15] and old-fashioned

expert systems such as Mycin [2]. A neural network for character recognition may be able to recognise an "A" from a bitmap, but could not say what an "A" looks like. In Mycin there are rules leading from the symptoms to the diseases, but the system can't tell you what the symptoms of some disease are.

- The second strategy is to model both causally and evidentially and to use the causal model for causal reasoning and the evidential model for evidential reasoning. The main problem with this is the redundancy of the knowledge, and its associated problem of consistency, although there are techniques for automatically inferring the evidential model from the causal model for limited cases [25, 7, 16, 29]. Pearl [23] has pointed out how naive representations of evidential and causal knowledge can lead to problems.

- The third strategy is to model causally and use different reasoning strategies for causal and evidential reasoning. For causal reasoning we can directly use the causal model, and for evidential reasoning we can use abduction.

This leads to an abstract formulation of abduction that will include both logical and probabilistic formulations of abduction:

Definition 1.4 Abduction is the use of a model in its opposite direction. That is, if a model specifies how x gives a y, abduction lets us infer x from y. Abduction is usually evidential reasoning from a causal model[3].

If we have a model of how causes produce effects, abduction lets us infer causes from effects. Abduction depends on an implicit assumption of complete knowledge of possible causes [7, 29]; when an effect is observed, one of its causes must be present.

1.2 Learning as an evidential reasoning task

In this section we explore learning as an evidential reasoning task. Given a task, a prior belief or bias, and some data, the learning task is to produce

[3]Neither the standard logical definition of abduction nor the probabilistic version of abduction (presented below) prescribe that the given knowledge is causal. It shouldn't be surprising that the formal definitions don't depend on the knowledge base being causal as the causal relationship is a modelling assumption. We don't want the logic to impose arbitrary restrictions on modelling.

an updated theory of the data (the posterior belief) that can be used in the task.

In order to make this clear, we must be very careful to distinguish:

- the task being learned
- the task of learning itself.

This distinction is very important when the task being learned is also an evidential reasoning task (e.g., learning to do diagnosis, or learning a perceptual task).

The task of learning can be seen as an evidential reasoning task where the model "causes" the data. The aim of learning is: given the data, to find appropriate models (evidential reasoning), and from the model(s) to make prediction on unseen cases (causal reasoning).

When we look at learning as an evidential reasoning task, not surprisingly, we find learning methods that correspond to the two strategies that allow causal and evidential reasoning (the second and third strategies of the previous section).

The second strategy of the previous section is to build special-purpose reasoning strategies to carry out the evidential reasoning task (i.e., inferring the model from the data) that is separate from the causal reasoning task (predicting new data from the model). Examples of such special purpose mechanisms are decision-tree learning algorithms such as C4.5 [33] and CART [1], and backpropagation for neural network learning [34].

The rest of this paper will show how the third strategy of the previous section, namely causal modelling and using different strategies for causal and evidential reasoning, can be used for learning, and can be carried out with both logical and probabilistic specifications of abductive reasoning. Such a strategy implies that we need a specification of the models to be learned and what these models predict in order to build a learning algorithm.

2 Bayesian Probability

In this section we introduce and motivate probability theory independently of learning. The interpretation of probability theory we use here is called *Bayesian*, *personal*, or *subjective* probability, as opposed to the *frequentist* interpretation of probability as the study of the frequency of repeatable events.

Probability theory [24] is a study of belief update; how an agent's knowledge affects its beliefs. An agent's probability of a proposition is a measure of how much the proposition is believed by the agent. Rather than considering an agent maintaining one coherent set of beliefs (for example, the most plausible way the world could be based on the agent's knowledge), Bayesian probability specifies that an agent must consider all possible ways that the world could be and their relative plausibilities. This plausibility when normalised to the range [0,1] so that the values for all possible situations sum to one is called a **probability**.

There are a number of reasons why we would be interested in probability, including:

- An agent can only act according to its beliefs and its goals. An agent doesn't have access to everything that is true in its domain, but only to its beliefs. An agent must somehow be able to decide on actions based on its beliefs.

- It is not enough for an agent to have just a single model of the world in which it is interacting and act on that model. It also needs to consider what other alternatives may be true, and make sure that its actions are not too disastrous if these other contingencies happen to arise.

  A classic example is wearing a seat belt; an agent may assume that it won't have an accident on a particular trip, but wears a seat belt to cover the possibility that it does have an accident. Under normal circumstances, the seat belt is a slight nuisance, but if there is an accident, the agent is much better off when it is wearing a seat belt. Whether the agent wears a seat belt depends on how inconvenient it is when there is no accident, how much better off the agent would be if they were wearing a seat belt when there is an accident, and how likely an accident is. This tradeoff between various outcomes, their relative desirability, and their likelihood is the subject of decision theory.

- As we will see below, probabilities are what can be obtained from data. Probability lets us explicitly model noise in data, and lets us update our beliefs based on noisy data.

The formalisation of probability theory is simple.

A **random variable** is a term in a language that can take one of a number of different values. The set of all possible values a variable can take

is called the **domain** of the variable. We write $x = v$ to mean the proposition that variable x has value v. A **Boolean random variable** is one where the domain is $\{true, false\}$. Often we write x rather than $x = true$ and $\neg x$ rather than $x = false$. A **proposition** is a Boolean formula made from assignments of values to variables.

Some example random variables may be a patient's blood pressure at 2:00p.m. on July 13, 1998, the value of the Australian dollar relative to the Canadian dollar on January 1, 2001, whether a patient has cancer at a particular time, whether a light is lit at some time point, or whether a particular coin lands heads on a particular toss.

There is nothing *random* about random variables. We introduce them because it is often useful to be able to refer to a variable without specifying its value.

Suppose we have a set of random variables. A **possible world** specifies an assignment of one value to each random variable. If w is a world, x is a random variable and v is a value in the domain of x, we write

$$w \models x = v$$

to mean that variable x is assigned value v in world w. We can allow Boolean combinations on the right-hand side of $\models$, where the logical connectives have their standard meaning, for example,

$$w \models \alpha \wedge \beta \text{ iff } w \models \alpha \text{ and } w \models \beta$$

So far this is just standard logic, but using the terminology of random variables.

Let's define a nonnegative measure $\mu(w)$ to each world w so that the measures of the possible worlds sum[4] to 1. The use of 1 is purely by convention; we could have just as easily used 100, for example.

The **probability** of proposition α, written $\mathbf{P}(\alpha)$, is the sum of the measures of the worlds in which α is true:

$$\mathbf{P}(\alpha) = \sum_{w \models \alpha} \mu(w).$$

The most important part of Bayesian probability is conditioning on observations. The set of all observations is called the **evidence**. If you are

[4]When there are infinitely many possible worlds, we need to use some form of measure theory, so that the measure of all of the possible worlds is 1. This requires us to assign probabilities to measurable sets of worlds, but the general idea is essentially the same.

given evidence e, conditioning means that all worlds in which e is false are eliminated, and the remaining worlds are renormalised so that their probabilities sum to 1. This can be seen as creating a new measure μ_e defined by:

$$\mu_e(w) = \begin{cases} 0 & \text{if } w \not\models e \\ \mu(w)/\mathbf{P}(e) & \text{if } w \models e \end{cases}$$

We can then define the **conditional probability** of a given e, written $\mathbf{P}(a|e)$ in terms of the new measure:

$$\mathbf{P}(a|e) = \sum_{w \models \alpha} \mu_e(w).$$

Example 2.1 The probability $\mathbf{P}(sneeze = yes|cold = severe)$ specifies, out of all of the worlds where *cold* is *severe*, what proportion have *sneeze* with value *yes*. It is the measure of belief in the proposition $sneeze = yes$ given that all you knew was that the cold was severe. The probability $\mathbf{P}(sneeze = yes|cold \neq severe)$ considers the other worlds where the cold isn't severe, and specifies the proportion of these in which *sneeze* has value *yes*. This second probability is independent of the first.

2.1 Bayes' Rule

Given the above semantic definition of conditioning, it is easy to prove:

$$\mathbf{P}(h|e) = \frac{\mathbf{P}(h \wedge e)}{\mathbf{P}(e)}.$$

Rewriting the above formula, and noticing that $h \wedge e$ is the same proposition as $e \wedge h$, we get:

$$\begin{aligned} \mathbf{P}(h \wedge e) &= \mathbf{P}(h|e) \times \mathbf{P}(e) \\ &= \mathbf{P}(e|h) \times \mathbf{P}(h) \end{aligned}$$

We can divide the right hand sides by $\mathbf{P}(e)$, giving

$$\mathbf{P}(h|e) = \frac{\mathbf{P}(e|h) \times \mathbf{P}(h)}{\mathbf{P}(e)}$$

if $\mathbf{P}(e) \neq 0$. This equation is known as **Bayes' theorem** or **Bayes' Rule**. It was first given in this generality by Laplace [17].

It may seem puzzling why such an innocuous looking equation should be so celebrated. It is important because it tells us how to do evidential reasoning from a causal knowledge base; **Bayes' rule is an equation for abduction**. Suppose $\mathbf{P}(e|h)$ specifies a causal model; it gives the propensity of effect e in the context when h is true. Bayes' rule specifies how to do evidential reasoning; it tells us how to infer the cause h from the effect e.

The numerator is the product of the likelihood, $\mathbf{P}(e|h)$, which specifies how well the hypothesis h predicts the evidence e, and the prior probability, $\mathbf{P}(h)$, that specifies how much the hypothesis was believed before any evidence arrived.

The denominator, $\mathbf{P}(e)$, is a normalising constant to ensure that the probabilities are well formed. If $\{h_1, \ldots, h_k\}$ are a set of pairwise incompatible (h_i and h_j cannot both be true if $i \neq j$) and covering (one h_i must be true) set of hypotheses, then

$$\mathbf{P}(e) = \sum_{h_i} \mathbf{P}(e|h_i) \times \mathbf{P}(h_i)$$

If you are only interested in comparing hypotheses this denominator can be ignored.

2.2 Bayesian Learning

Bayesian learning, or Bayesian statistics [5, 19, 12, 13] is the method for using Bayes' rule for evidential reasoning for the evidential reasoning task of learning.

Bayes' rule is

$$\mathbf{P}(h|e) = \frac{\mathbf{P}(e|h) \times \mathbf{P}(h)}{\mathbf{P}(e)}.$$

If e is the data (all of the training examples), and h is a hypothesis, Bayes' rule specifies how, given the model of how the hypothesis h produces the data e and the prior propensity of h, you can infer how likely the hypothesis is, given the data.

One of the main reasons why this is of interest is that the hypotheses can be noisy; an hypothesis can specify a probability distribution over the data it predicts. Moreover, Bayes' rule allows us to compare those hypotheses that predict the data exactly (where $\mathbf{P}(e|h) = 1$) amongst themselves and with the hypotheses that specify any other probability of the data.

Example 2.2 Suppose we are doing Bayesian learning of decision trees, and are considering a number of definitive decision trees (i.e., they predict classifications with 0 or 1 probabilities, and thus have no room for noise). For each such decision tree h, either $\mathbf{P}(e|h) = 1$ or $\mathbf{P}(e|h) = 0$. Bayes theorem tells us that those that don't predict the data have posterior probability 0, and those that predict the observed data have posterior probabilities proportional to their priors. Thus the prior probability specifies the learning bias (for example, towards simpler decision trees); out of all of the trees that match the data, which are to be preferred. Without such a bias, there can be no learning as every possible function can be represented as a decision tree. Bayes rule also specifies how to compare simpler decision trees that may not exactly fit the data (e.g., if they have probabilities at the leaves) with more complex ones that exactly fit the data. This gives a principled way to handle overfitting.

Example 2.3 The simplest form of Bayesian learning with probabilistic hypotheses is when there is a single binary event that is repeated and statistics are collected. That is, we are trying to learn probabilities. Suppose we have some object that can fall down such that either there is some distinguishing feature (which we will call *heads*) showing on top, or there is not heads (which we will call *tails*) showing on top. We would like to learn the probability that there is a heads showing on top. Suppose our hypothesis space consists of hypotheses that specify $\mathbf{P}(heads) = p$ where *heads* is the proposition that says heads is on top, and p is a number that specifies the probability of a heads on top. Implicit in this hypothesis is that repeated tosses are independent[5]. Suppose we have on observation e consisting of a particular sequence of outcomes with n outcomes with *heads* true and out of m outcomes. Let h_p be the hypothesis that $\mathbf{P}(heads) = p$ for some $0 \leq p \leq 1$. Then we have, by elementary probability theory,

$$\mathbf{P}(e|h_p) = p^n(1-p)^{m-n}$$

Suppose that our prior probability is uniform on [0,1]. That is, we consider each value for $\mathbf{P}(heads)$ to be equally likely before we see any data.

Figure 2 shows the posterior distributions for various values of n and m. Note that the only hypotheses that are inconsistent with the observations

[5]Bayesian probability doesn't require independent trials. You can model the interdependence of the trials in the hypothesis space.

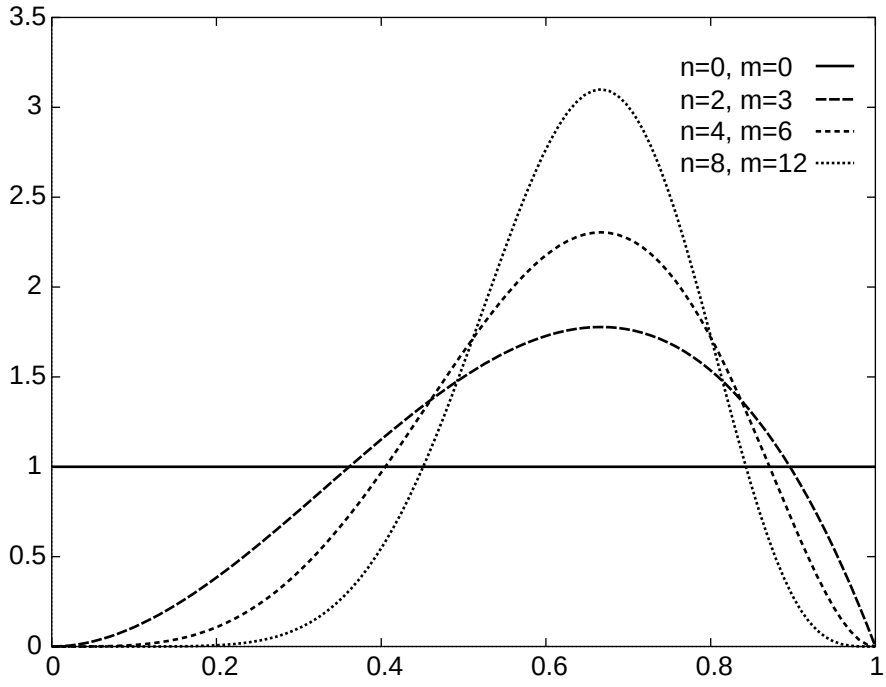

Figure 2: Posterior distribution for learning a probability

are $\mathbf{P}(heads) = 0$ when $n > 0$ and $\mathbf{P}(heads) = 1$ when $m > 0$. Note that if the prior isn't very biased, it soon gets dominated by the data.

Bayesian learning has been applied to many representations including decision trees [3], neural networks [21], Bayesian networks [10], and unsupervised learning [6]. All we need is a way to specify what a particular decision tree, neural network, Bayesian network, or logic program predicts (this is well defined by the definition of the representation), as well as a prior probability on the different representations.

Prior probabilities may seem to be problematic, but are important for avoiding overfitting. They give a principled way to do what would otherwise have to be done by some ad hoc mechanism, such as pruning decision trees or limiting the size of neural networks. For example, if there is noise in the data, a more detailed decision tree can always be made to fit the data better, but usually has worse predictive properties on unseen examples. A prior probability on decision trees provides a bias that lets us tradeoff fitting the training data with simplicity of the trees [3].

Bayesian leaning is closely related to the minimum description length (MDL) principle. If we were to choose the most likely hypothesis given the data[6] (called the maximum a posteriori probability, or MAP, hypothesis), we can use:

$$\begin{aligned}
&\arg\max_h \mathbf{P}(h|e) \\
&= \arg\max_h \frac{\mathbf{P}(e|h) \times \mathbf{P}(h)}{\mathbf{P}(e)} \\
&= \arg\max_h \mathbf{P}(e|h) \times \mathbf{P}(h) \\
&= \arg\max_h -\log_2 \mathbf{P}(e|h) + -\log_2 \mathbf{P}(h)
\end{aligned}$$

The latter is the number of bits it takes to describe the data in terms of the model plus the number of bits it takes to describe the model. Thus the best hypothesis is the one that gives the shortest description of the data in terms of that model.

[6] We don't have to do this. In particular, it is the posterior distribution of the hypotheses that we want to use to make decisions, rather than the most likely hypothesis.

3 Bayesian Networks

Probability specifies a semantic construction and not a representation of knowledge. A Bayesian network [24] is a way to represent probabilistic knowledge. The idea is to represent a domain in terms of random variables and to explicitly model the interdependence of the random variables in terms of a graph. This is useful when a random variable only depends on a few other random variables, as occurs in many domains.

Suppose we decide to represent some domain using the random variables $x_1, \ldots, x_n$. If we totally order the variables, it is easy to prove that

$$\begin{aligned}&\mathbf{P}(x_1, \ldots, x_n)\\ &\quad = \mathbf{P}(x_1)\mathbf{P}(x_2|x_1)\mathbf{P}(x_3|x_1, x_2)\cdots\mathbf{P}(x_n|x_1\cdots x_{n-1})\end{aligned}$$

For each variable x_i suppose there is some minimal set $\pi_{x_i} \subseteq \{x_1, \ldots, x_{i-1}\}$ such that

$$\mathbf{P}(x_i|x_1, \ldots, x_{i-1}) = \mathbf{P}(x_i|\pi_{x_i})$$

That is, once you know the values of the variables in π_{x_i}, knowing the values of other predecessors of x_i in the total ordering will not change your belief in x_i. The elements of the set π_{x_i} are known as the **parents** of variable x_i. We say x_i is **conditionally independent** of its predecessors given its parents. We can create a graph where there is an arc from each parent of a node into that node. Such a graph, together with the conditional probabilities for $\mathbf{P}(x_i|\pi_{x_i})$ for each variable x_i is known as a **Bayesian network** or a **belief network** [24, 14].

There are a few important points to notice about a Bayesian network:

- By construction, the graph defining a Bayesian network is acyclic.
- Different total orderings of the variables can result in different Bayesian networks for the same underlying distribution.
- The size of the conditional probability table $\mathbf{P}(x_i|\pi_{x_i})$ is exponential in the number of parents of x_i.

Typically we try to build Bayesian networks so that the total ordering implies few parents and a sparse graph.

Bayesian networks are of interest because they can be constructed taking into account just local information, the information that has to be specified

is reasonably intuitive, and there are many domains that have concise representations as Bayesian networks. There are algorithms that can exploit the sparseness of the graph for computational gain [18, 9, 36], exploit the skewness of distributions [30] or use the structure for stochastic simulation [11, 22, 8].

4 Bayesian learning and logic-based abduction

So far we have given an informal characterisation of Bayes' rule as a rule for abduction. Poole [28] has shown a direct correspondence between Bayesian networks and logic-based conceptions of abduction. Buntine [4] has shown how Bayesian networks form a representation for many inductive learning tasks. In this section we put these together to show how inductive learning tasks can be related to logic-based abduction. In the following section, we expand on this mapping to discuss some of the issues of this book relating abduction and induction.

4.1 Logic Programs, Abduction and Bayesian Networks

This section overviews the relationship between Bayesian networks and logic-based abduction [28]. In particular, I give the translation of Bayesian networks into probabilistic Horn abduction [28], a form of probabilistic logic programs.

Suppose variable a has parents $b_1, \ldots, b_k$ in a Bayesian network. As part of the Bayesian network are probabilities of the form

$$\mathbf{P}(a = v | b_1 = v_1, \ldots, b_k = v_k) = p$$

These can be translated into rules of the form:

$$a = v \quad \leftarrow \quad b_1 = v_1 \wedge \ldots \wedge b_k = v_k \wedge h. \tag{1}$$

which can be treated as normal logical rules where h is assumable.

In probabilistic Horn abduction (and its successor the independent choice logic [31], which can handle more general rules, including negation as failure, as well as different agents choosing assumptions), the assumables are structured in terms of a **choice space**, $\mathbf{C}$, which is a set of alternatives (called

disjoint sets in [28]), where an **alternative** is a set of ground atoms. Each member of an alternative is assumable and can only appear in one alternative. The integrity constraints are that the elements of an alternative are pairwise inconsistent.

An independent choice logic theory is specified by a choice space and an acyclic logic program that doesn't imply any element of an alternative. The semantics is defined in terms of possible worlds. There is a possible world for each selection of one element from each alternative. What is true in a possible world is given by the stable model of the logic program and the atoms selected. The logic is abductive in the sense that the explanations of g form a concise specification of the possible worlds in which g is true [28, 32].

We place a probability over the assumables so that the probability of the elements of an alternative sum to one. We assume that the different alternatives are probabilistically independent (the alternatives correspond to random variables).

In term of representing the Bayesian network above, there is an alternative for each assignment of values to the parents of a. For each such alternative, there is an element of the alternative for each value of a. The probability of the assumable h (from equation (1)) is the same as the probability of the corresponding conditional probability in the Bayesian network:

$$\mathbf{P}(h) = \mathbf{P}(a = v | b_1 = v_1, \ldots, b_k = v_k)$$

The abductive characterisation of probabilistic Horn abduction is straightforward. For any proposition h, the probability of h can be computed from the set of minimal explanations of h. The minimal explanations are disjoint (by the way the rules were constructed), and so the probability of h is the sum of the probabilities of the minimal explanations for h. The probability of an explanation is the product of the probabilities of the assumables. That is

$$\mathbf{P}(h) = \sum_{e \text{ is a minimal explanation of } h} \mathbf{P}(e)$$

where the probability for explanation e is given by

$$\mathbf{P}(e) = \prod_{n \in e} \mathbf{P}(n)$$

In [28] it was proved that the Bayesian network and the abductive characterisation result in the same probabilities.

Suppose we want to compute a probability given evidence, we have

$$\mathbf{P}(h|e) = \frac{\mathbf{P}(h \wedge e)}{\mathbf{P}(e)}$$

Thus this can be seen in terms of abduction as: given evidence e, first explain the evidence (this gives $\mathbf{P}(e)$), and from the explanations of the evidence, explain h (this gives $\mathbf{P}(h \wedge e)$). Note that the explanation of $h \wedge e$ are the explanations of e extended to also explain h. In terms of a Bayesian network, you can first go backwards along the arrows to explain the evidence, and then go forward along the arrows to make predictions. Thus not only can Bayes' rule be seen as a rule for abduction, but Bayesian networks can be seen a representation for abduction. Note that this reasoning framework of using abduction for evidential reasoning and assumption-based reasoning for causal reasoning (see Figure 1), which is what the above analysis gives us for Bayesian networks, has also been proposed in the default reasoning literature [26, 27, 35].

The logic programs have a standard logical meaning and can be extended to include (universally quantified) logical variables[7] in the usual way. The only difference to standard logic programs[8] is that some of the premises are hypotheses that may have an associated probability.

4.2 Bayesian networks and induction

Buntine [4] argues that Bayesian networks (as well as related chain graphs) form a good representation for many induction tasks. That is, he argued that Bayesian networks can form a representation for the evidential reasoning task of learning.

Note that this is very different from the problem of learning Bayesian networks themselves for which there are Bayesian and non-Bayesian techniques (see [10] for a review of learning Bayesian networks). Buntine was using Bayesian networks to represent the task of learning, independently of the task being learned.

[7] It is important not to confuse logical variables, which stand for individuals, and random variables. In this paper, I will follow the Prolog convention of having logical variables in upper case.

[8] In the independent choice logic [31], we can also have negation as failure in the rules. The notion of abduction needs to be expanded to allow abduction through the negation [32].

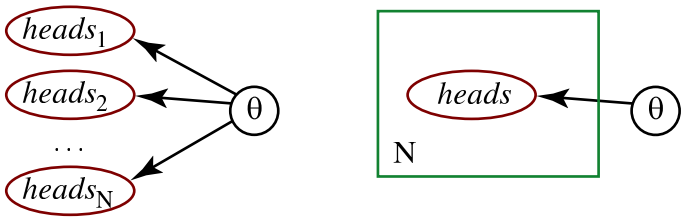

Figure 3: Bayesian network for coin tossing, with and without plates

Buntine used the notion of *plates* which were repeated copies of a network.

Example 4.1 Figure 3 shows a Bayesian network for the coin tossing of Example 2.3. The probability of heads on example i, which in the left-hand side of Figure 3 is shown as $heads_i$, is a random variable that depends only on θ, the probability of heads appearing on a coin toss. The right-hand side of Figure 3 shows the same network using plates, where there is one copy of the boxed node for each example.

Given the logic-programming characterisation of Bayesian networks, we can use universally quantified logical variables in the rules to represent the plates of Buntine.

Example 4.2 Let's write the example of Figure 3 in terms of probabilistic Horn abduction. First we can represent each arc to an example as the rule:

$$heads(E) \leftarrow happens\_to\_turn\_heads(E, P) \wedge prob\_of\_heads(P)$$

$$tails(E) \leftarrow happens\_to\_turn\_tails(E, P) \wedge prob\_of\_heads(P)$$

where $heads(E)$ is true if example E shows a heads, and $tails(E)$ is true if example E shows a tails.

The corresponding alternatives are

$$\forall E \forall P \{happens\_to\_turn\_heads(E, P), happens\_to\_turn\_heads(E, P)\} \in \mathbf{C}$$

That is, we can assume that example E turns heads or assume it turns tails. We then have the probabilities:

$$\mathbf{P}(happens\_to\_turn\_heads(E, P)) = P$$

$$\mathbf{P}(happens\_to\_turn\_tails(E, P)) = 1 - P$$

We also have the alternative that corresponds to the θ in Figure 3:

$$\{prob\_of\_heads(P) : \ 0 \leq P \leq 1\} \in \mathbf{C}$$

That is you can assume any single probability in the range $[0, 1]$.

Suppose you have example $e_1, \ldots, e_k$, and have observed say

$$heads(e_1), tails(e_2), tails(e_3), \ldots$$

The explanations of this observation are of the form:

$$\{happens\_to\_turn\_heads(e_1, P), happens\_to\_turn\_tails(e_2, P),$$
$$happens\_to\_turn\_tails(e_3, P), \ldots,$$
$$prob\_of\_heads(P)\}$$

for each $P \in [0, 1]$. Suppose there were n heads and m tails in the $k = n + m$ examples, then the probability of this explanation is

$$P^n \times (1 - P)^m \times q$$

where q is $\mathbf{P}(prob\_of\_heads(P))$.

5 Combining induction and abduction

In terms of abduction, the basic idea of this model of induction is to have some assumptions that are specific to each example, and some assumptions that are specific to the model being learned. For each example, you make some model-specific assumptions and some example-specific assumptions (that also depend on the model assumptions). When explaining a number of examples, they each have their own example-specific assumptions, but must share the model assumptions.

Buntine [4] has shown how many different learning algorithms from neural networks to unsupervised learning can be put into this framework.

5.1 Learning decision trees

In this section we will sketch how the same framework can be used for more complicated examples, where the models must be constructed, rather than having a fixed number of parameters to be estimated. Here the flexibility of representation in terms of logic-based abduction can be seen to have great advantages over the use of plates [4].

Let's look at the same framework for Bayesian learning of decision trees with probabilities at the leaves[9] [3]. To keep this simple let's suppose that all attributes are Boolean.

We use the relation $prop(Ex, Att, Val)$ that is true when example Ex has value Val on attribute Att. Suppose a decision tree is either a number or is of the form $if(C, YT, NT)$ where C is an attribute YT and NT are trees.

We need to write rules that specify the value of the classification based on the tree:

$$prop(Ex, classification, V) \leftarrow tree(T) \wedge tree\_predicts(T, Ex, V).$$

It is straightforward to define what a tree predicts:

$$\begin{aligned}
&tree_predicts(if(C, YesT, NoT), Ex, V) \leftarrow \\
&\quad prop(Ex, C, true) \wedge \\
&\quad tree_predicts(YesT, Ex, V). \\
&tree_predicts(if(C, YesT, NoT), Ex, V) \leftarrow \\
&\quad prop(Ex, C, false) \wedge \\
&\quad tree_predicts(NoT, Ex, V). \\
&tree_predicts(N, Ex, V) \leftarrow \\
&\quad number(N) \wedge \\
&\quad predicts_prob(Ex, N, V).
\end{aligned}$$

where

$$\forall Ex \forall N \{predicts\_prob(Ex, N, true), predicts\_prob(Ex, N, false)\} \in \mathbf{C}$$

such that

$$\begin{aligned}
&\mathbf{P}(predicts_prob(Ex, N, true)) = N \\
&\mathbf{P}(predicts_prob(Ex, N, false)) = 1 - N
\end{aligned}$$

[9]Note that when these decision trees are translated into rules, probabilistic Horn abduction theories result. But here we are using probabilistic Horn abduction to represent the learning task, not the task being learned.

Similarly we need ways to abduce what the trees are, and (the more difficult) problem of assigning the priors on the decision trees.

The most likely explanation of a set of classifications on examples results in the most likely decision tree given those examples.

5.2 Generalization

It has often been thought that probability is unsuitable for generalization as the generalization $\forall X\ r(X)$ must have a lower probability than any set of examples $r(e_1), \ldots, r(e_k)$, as the generalization implies the examples. While the statement of probability is correct, it is misleading because it is not the hypothesis and the evidence that we want to compare but the different hypotheses[10].

The different hypotheses may be, for example:

1. $r(X)$ is always true,
2. $r(X)$ is sometimes true (and it just happened to be true for examples $e_1, \ldots, e_k$).
3. $r(X)$ is always false.

This can be represented as having the alternatives:

$$\{r\_always\_true, r\_sometimes\_true, r\_always\_false\} \in \mathbf{C}$$
$$\forall X \{r\_happens\_true(X), r\_happens\_false(X)\} \in \mathbf{C}$$

with some probabilities associated with the assumables, and the rules

$$r(X) \leftarrow r\_always\_true.$$
$$r(X) \leftarrow r\_sometimes\_true \wedge r\_happens\_true(X).$$

For any set of (all positive) observations: $r(e_1), \ldots, r(e_k)$, there are two competing explanations:

$$\{r\_always\_true\}$$
$$\{r\_sometimes\_true, r\_happens\_true(e_1), \ldots, r\_happens\_true(e_k)\}$$

[10] It is interesting to note that in the abductive framework the hypothesis *always* implies the evidence, and so it is always less likely. But this is exactly what we want from learning: we want the learned hypothesis to make risky prediction, that could be wrong, on unseen data.

If there are no extreme (0 or 1) probabilities, with enough positive examples, the conclusion that r is always true will be the most likely hypothesis. Thus we can make universal generalizations within this framework.

6 Conclusion

This paper has related the Bayesian approach to learning with logic-based abduction. In particular, I have sketched the the relationship between Bayesian leaning and the graphical models of Buntine [4] and the relationship between graphical models and abductive logic programming of Poole [28]. It should be emphasised that, while each of the links has been developed, the chain has not been fully investigated. This paper should be seen as a starting point, rather than a survey of mature work.

References

[1] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees.* Wadsworth and Brooks, Monterey, CA, 1984.

[2] B. Buchanann and E. Shortliffe, editors. *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project.* Addison-Wesley, Reading, MA, 1984.

[3] W. Buntine. Learning classification trees. *Statistics and Computing*, 2:63–73, 1992.

[4] W. L. Buntine. Operations for learning with graphical models. *Journal of Artificial Intelligence Research*, 2:159–225, 1994.

[5] P. Cheeseman. On finding the most probable model. In J. Shranger and P. Langley, editors, *Computational Models of Scientific Discovery and Theory Formation*, chapter 3, pages 73–95. Morgan Kaufmann, San Mateo, CA, 1990.

[6] P. Cheeseman, J. Kelly, M. Self, J. Stutz, W. Taylor, and D. Freeman. Autoclass: A Bayesian classification system. In *Proc. Fifth International Conference on Machine Learning*, pages 54–64, Ann Arbor, MI, 1988.

[7] L. Console, D. Theseider Dupré, and P. Torasso. Abductive reasoning through direct deduction from completed domain models. In W. R. Zbigniew, editor, *Methodologies for Intelligent Systems 4*, pages 175–182. Elsiever Science Publishing Co., 1989.

[8] P. Dagum and M. Luby. An optimal approximation algorithm for Bayesian inference. *Artificial Intelligence*, 93(1–2):1–27, 1997.

[9] R. Dechter. Bucket elimination: A unifying framework for probabilistic inference. In E. Horvitz and F. Jensen, editors, *Proc. Twelfth Conf. on Uncertainty in Artificial Intelligence (UAI-96)*, pages 211–219, Portland, OR, 1996.

[10] D. Heckerman. A tutorial on learning with Bayesian networks. Technical Report MSR-TR-95-06, Microsoft Research, March 1995. (Revised November 1996).

[11] M. Henrion. Propagating uncertainty in Bayesian networks by probabilistic logic sampling. In J. F. Lemmer and L. N. Kanal, editors, *Uncertainty in Artificial Intelligence 2*, pages 149–163. Elsevier Science Publishers B.V., 1988.

[12] E.T. Jaynes. Bayesian methods: General background. In J.H. Justice, editor, *Maximum Entropy and Bayesian Methods in Applied Statistics*, pages 1–25. Cambridge University Press, Cambridge, England, 1985.

[13] E.T. Jaynes. *Probability Theory: The Logic of Science*. Unpublished Manuscript, 1995. Available from `ftp://bayes.wustl.edu/Jaynes.book`.

[14] F. V. Jensen. *An Introduction to Bayesian Networks*. Springer Verlag, New York, 1996.

[15] M.I. Jordan and C. Bishop. Neural networks. Memo 1562, MIT Artificial Intelligence Lab, Cambridge, MA, March 1996.

[16] K. Konolige. Abduction versus closure in causal theories. *Artificial Intelligence*, 53(2-3):255–272, 1992.

[17] P.S. Laplace. *Théorie Analytique de Probabilités*. Courcier, Paris, 1812.

[18] S. L. Lauritzen and D. J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society, Series B*, 50(2):157–224, 1988.

[19] T.J. Loredo. From Laplace to supernova SN 1987A: Bayesian inference in astrophysics. In P.F. Fougère, editor, *Maximum Entropy and Bayesian Methods*, pages 81–142. Kluwer Academic Press, Dordrecht, The Netherlands, 1990.

[20] A. K. Mackworth. Vision research strategy: black magic, metaphors, mechanisms, miniworlds and maps. In A. R. Hanson and E. M. Riseman, editors, *Computer Vision Systems*, pages 53–61. Academic Press, New York, NY, 1978.

[21] R.M. Neal. *Bayesian Learning for Neural Networks*. Springer-Verlag, New York, 1996.

[22] J. Pearl. Evidential reasoning using stochastic simulation of causal models. *Artificial Intelligence*, 32(2):245–257, May 1987.

[23] J. Pearl. Embracing causation in default reasoning. *Artificial Intelligence*, 35(2):259–271, 1988.

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

[25] D. Poole. Representing knowledge for logic-based diagnosis. In *International Conference on Fifth Generation Computing Systems*, pages 1282–1290, Tokyo, Japan, November 1988.

[26] D. Poole. Explanation and prediction: An architecture for default and abductive reasoning. *Computational Intelligence*, 5(2):97–110, 1989.

[27] D. Poole. A methodology for using a default and abductive reasoning system. *International Journal of Intelligent Systems*, 5(5):521–548, December 1990.

[28] D. Poole. Probabilistic Horn abduction and Bayesian networks. *Artificial Intelligence*, 64(1):81–129, 1993.

[29] D. Poole. Representing diagnosis knowledge. *Annals of Mathematics and Artificial Intelligence*, 11:33–50, 1994.

[30] D. Poole. Probabilistic conflicts in a search algorithm for estimating posterior probabilities in Bayesian networks. *Artificial Intelligence*, 88:69–100, 1996.

[31] D. Poole. The independent choice logic for modelling multiple agents under uncertainty. *Artificial Intelligence*, 94:7–56, 1997. Special issue on economic principles of multi-agent systems.

[32] D. Poole. Abducing through negation as failure: stable models in the Independent Choice Logic. *Journal of Logic Programming*, to appear, 1998. Available from `http://www.cs.ubc.ca/spider/poole/abstracts/approx-pa.html`.

[33] J. R. Quinlan. *C4.5 Programs for Machine Learning*. Morgan Kaufmann, San Mateo, CA, 1993.

[34] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. In D. E. Rumelhart and J. L. McClelland, editors, *Parallel Distributed Processing*, chapter 8, pages 318–362. MIT Press, Cambridge, MA, 1986.

[35] M. Shanahan. Prediction is deduction, but explanation is abduction. In *Proc. 11th International Joint Conf. on Artificial Intelligence (IJCAI-89)*, pages 1055–1060, Detroit, MI, August 1989.

[36] N.L. Zhang and D. Poole. Exploiting causal independence in Bayesian network inference. *Journal of Artificial Intelligence Research*, 5:301–328, December 1996.