

Logical Argumentation, Abduction and Bayesian Decision Theory: A Bayesian Approach to Logical Arguments and it's Application to Legal Evidential Reasoning

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

April 7, 2000

Abstract

There are good normative arguments for using Bayesian decision theory for deciding what to do. However, there are also good arguments for using logic, where we want have a formal semantics for a language and use the structure of logical argumentation with logical variables to represent multiple individuals (things). This paper shows how decision theory and logical argumentation can be combined into a coherent framework. The Independent Choice Logic can be viewed as first-order representation of belief networks with conditional probability tables represented as first-order rules, or as a abductive/argument-based logic with probabilities over assumables. Intuitively we can use logic to model causally (in terms of logic programs with assumables). Given evidence, we abduce to the explanations, and then can predict what follows from these explanations. As well as abduction to the best explanation(s), from which we can bound probabilities, we can also do marginalization to reduce the detail of arguments. An example of Tillers is given is used to show the how the framework could be used for legal reasoning. The code to run this example is available from the authors web site.

1 Introduction

There are good normative arguments for using logic to represent knowledge (Nilsson, 1991; Poole, Mackworth & Goebel, 1998). These arguments are usually based on reasoning with symbols with an explicit denotation, allowing relations amongst individuals, and permitting quantification over individuals. This is often translated as needing (at least) the first-order predicate calculus. Unfortunately, the first-order predicate calculus has very primitive mechanisms for handling uncertainty, namely, the use of disjunction and existential quantification.

There are also good normative reasons for using Bayesian decision theory for decision making under uncertainty (Von Neumann & Morgenstern, 1953; Savage, 1972). These arguments can be intuitively interpreted as seeing decision making as a form of gambling, and that probability and utility are the appropriate calculi for gambling.

It is important to note that decision theory has nothing to say about representations. Adopting decision theory doesn't mean adopting any particular representation. While there are some representations that can be directly extracted from the theory, such as the explicit reasoning over the state space or the use of decision trees, these become intractable as the problem domains become large; it is like theorem proving by enumerating the interpretations. Adopting logic doesn't mean you have to enumerate interpretations, nor does adopting decision theory mean you have to use analogous representations.

The independent choice logic can be seen as a representation that combines logic and Bayesian decision theory.

First, I will talk about knowledge representation, in which tradition this representation is built. The ICL will then be presented from three alternate viewpoints: as a semantic framework in terms of choices made by agents, in terms of first-order belief networks (Bayesian networks) and as a framework for abduction and argumentation. I will then show some axioms from an example of Tillers, and show the outputs of our prototype implementation.

1.1 Knowledge Representation

In order to understand what AI can bring to table, Figure 1 (from (Poole et al., 1998)) shows the knowledge representation (KR) view. Given a problem we want a solution to, we find a representation for the problem, which we can compute to find an answer that can be interpreted as a solution to the problem.

When considering representations, there are a number of often competing considerations:

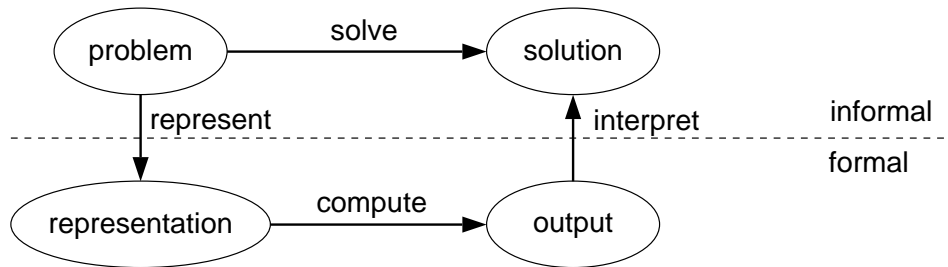


Figure 1: Knowledge Representation Framework

- The representation should be rich enough to be able to contain enough information to actually solve the problem.
- The representation should be as close to the problem as possible. We want the representation to be as “natural” as possible, so that a small changes in the problem result in small changes in the representation. Ideally it should be clear what knowledge is expressed so that we can directly argue about the correctness of the knowledge expressed in the representation.
- We want the representation to be amenable to efficient computation. This does not necessarily mean that the representation needs to be efficient in the worst case (because that usually invalidates the first consideration). Rather we would like to be able to exploit features of the problem for computational gain. This means that the representation must be capable of expressing those features of the problem that can be exploited computationally.

Belief networks (or Bayesian networks) (Pearl, 1988) are of interest because they provide a language that is represents the sort of knowledge a person may have about a domain, is rich enough for many applications and because features of the representation can be exploited for computational gain.

Unfortunately, the underlying logic is propositional. We cannot have relations amongst individuals as we can, for example, in the first-order predicate calculus. The predicate calculus, however has only primitive mechanisms for handling uncertainty (disjunction and existential quantification).

2 The Independent Choice Logic

The independent choice logic (ICL) is a knowledge representation that can be seen in a number of different ways:

- It is a way to add Bayesian probability to the predicate logic. In particular we want to have all uncertainty to be handled by probabilities (of for decision problems, as choices of various agents). So we start with logic programs, which can be seen as predicate logic with no uncertainty (no disjunctive assertions), and have independent choices that have associated probability distributions. The logic program specifies what follows from the choices made.
- It is a way to lift belief networks into a first-order language. In particular a belief network can be seen as a deterministic system with noise inputs (Pearl, 1999; Pearl, 2000). The deterministic system is modelled as a logic program. This can be seen as writing the conditional probability tables in rule form (which also naturally expresses context-specific independence). The noise inputs are given in terms of independent choices.
- It is a sound way to have probabilities over assumptions. Explaining observations means that we use abduction; we find the explanations (set of hypotheses) that imply the observations, and from these we make predictions. This reasoning is sound probabilistic inference.

2.1 Formal Semantics

In this section we give the language and the semantics of the ICL. This is simplified slightly; the general ICL allows for negation as failure in the logic programs (Poole, 2000) and choices by various agents (Poole, 1997) which lets us model decisions in a decision-theoretic or game-theoretic situation.

We assume that we have atomic formulae as in a normal logical language. We use the Prolog convention of having variables in upper case, and predicate symbol and function symbols in lower case.

A **clause** is either an atom or is of the form

$$h \leftarrow a_1 \wedge \cdots \wedge a_k$$

where h is an atom and each a_i is an atom. If $k = 0$ we just write h . All of the variables are assumed to be universally quantified in the scope of the clause.

A **logic program** is a set of clauses. We assume the logic program is acyclic¹.

An **atomic choice** can be any atom that does not unify with the head of any clause. An **alternative** is a set of atomic choices. A **choice space** is a set of alternatives such that an atomic choice can be in at most one alternative.

An ICL theory consists of

¹All recursions for variable-free queries eventually halt. We disallow programs such as $\{a \leftarrow a\}$ and $\{a \leftarrow b, b \leftarrow a\}$

F the facts, an acyclic logic program

C a choice space

P_0 a probability distribution over the alternatives in C . That is $P_0 : \cup C \rightarrow [0, 1]$ such that

$$\forall A \in C \sum_{c \in A} P_0(c) = 1$$

The semantics is defined in terms of possible worlds, and a probability distribution over possible worlds. Here we present the semantics for the case of a finite choice space, where there are only finitely many possible worlds. The more general case is considered in other places (Poole, 1997; Poole, 2000).

A **total choice** for choice space C is a selection of exactly one atomic choice from each alternative in C .

There is a **possible world** for each total choice. What is **true** in a possible world is defined by the atoms chosen by the total choice together with the logic program. (The acyclicity guarantees there there is a single model for each possible world). The probability of a possible world is the product of the values $P_0(c)$ for each c selected by the atomic choices.

The probability of a proposition is the sum of the probability of each possible world in which the proposition is true.

2.2 ICL and Belief networks

It may seem that, with independent alternatives, that the ICL is restricted in what it can represent. This is not the case; in particular it can represent anything the is representable by a Belief network. Moreover the translation is local, and (if all alternatives are binary) there is the same number of alternatives as there are free parameters in the belief network.

For example, if we had binary variables A , B and C , with domains $\{a, \neg a\}$, $\{b, \neg b\}$ and $\{c, \neg c\}$, where B and C are the parents of A , we will have rules such as

$$a \leftarrow b \wedge \neg c \wedge aifbnc$$

where $aifbnc$ is an atomic choice where $P_0(aifbnc)$ has the same value as the conditional probability as $P(a|b, \neg c)$ in the belief network. This generalizes to arbitrary discrete belief networks in the analogous way (Poole, 1993b).

This representation lets us naturally specify context-specific independence (Poole, 1997), where, for example, A may be independent of C when B has value $\neg b$ but is dependent when B has value b .

More importantly, this mapping lets us see the relationship of belief networks to logical languages. The logic programs are standard logic programs (they can even have negation as failure (Poole, 2000)). Viewing belief networks as logic programs gives us a natural way to lift them to the first-order case (i.e., with logical variables universally quantified over individuals).

2.3 ICL, Abduction and Logical Argumentation

The ICL can also be seen as a language for abduction. In particular, if all of the atomic choices are assumable (they are abducibles or possible hypotheses). An **explanation** for g is a consistent set of atomic choices that implies g . Consistency means that there is no more than one atomic choice from any alternative. An explanation can be seen as an argument based on explicit assumptions about what is true.

Each of these explanations has an associated probability obtained by computing the product of the probabilities of the atomic choices that make up the explanation. The probability of g can be computed by summing² the probabilities of the explanations for g (Poole, 1993b; Poole, 2000).

If we want to do evidential reasoning and compute $P(g|obs)$, we notice that this is $P(g \wedge obs)/P(obs)$. In terms of explanations, we can first find the explanations for obs (which would give us $P(obs)$) and then try to extend these explanations to also explain g (this will give us $P(g \wedge obs)$). Intuitively, we explain all of the explanations and see what these also predict.

We can also bound the prior and posterior probabilities by generating only a few of the most plausible explanations (either top-down (Poole, 1993a) or bottom-up (Poole, 1996)). Thus we can go inference to the best explanations to do sound (approximate) probabilistic reasoning.

2.4 Reasoning in the ICL

To do reasoning in the ICL we can either do

- variable elimination (marginalization or partial evaluation) to simplify the model (Poole, 1997). We sum out variables to reduce the detail of the representation.

²This assumes the bodies for the rules for each atom a are mutually exclusive. This is a common practice in logic programming and the rules obtained from the translation from belief networks have this property. We need to do something a bit more sophisticated if the rules are not disjoint (Poole, 2000).

- Generating explanations to bound the probabilities. Note that if we generate all of the explanations we could compute the probabilities exactly, but there are combinatorially many explanations.

In the description of what follows, I only do the second.

3 Tillers' Example

Peter Tillers presented an example of judicial proof in the quandary of Able Attorney. In this section I present a representation of part of this example to show how the ICL could be used for this sort of problem. A complete listing of the ICL representation is given below. The fixed width font gives the actual input.

All of the background knowledge is given in the form of rules and alternatives. The particulars of the case is given in terms of the observations.

Note also that the axiomatization is clumsy in a number of areas. In particular the ICL provides no facilities for dealing with time (all of the times in the story are ignored), modalities (e.g., reasoning about saying, believing, thinking, possibility, obligation, etc.), aggregates (e.g., probabilities that deal with reasoning about population sizes), dealing with language (e.g., understanding what a vicious SOB may be).

Finally the numbers are arbitrary³ and the actual rules are pretty stupid. This axiomatization is only intended to give an idea of what can be done.

3.1 Observations

Before we give the axiomatization, the observations that we deal with are:

- *says(peter, wentto(peter, hvstore))*
Peter says that he went to the Happy Valley Store.
- *says(peter, clerk_at(harry, hvstore))*
Peter says that Harry was a clerk at the Happy Valley Store
- *says(peter, vicious_sob(harry))*
Peter says that Harry is a vicious SOB.
- *says(peter, observed(peter, blinding_flash))*
Peter says that he observed a blinding flash.

³I tuned them a bit because in my first attempts it was always much more likely that someone was lying than they were truthful when they said some unlikely event occurred.

- `says(peter, says(doctor, shot(peter)))`
Peter said that the doctor said he was shot.
- `says(peter, says(newspaper, disappeared(harry)))`
Peter said that the newspaper said Harry disappeared.

We ask to explain the conjunction of these to determine the most likely explanations and to derive conditional probabilities.

3.2 Witness Honesty

The first collection of clauses specifies why someone may say something. We divide the people into honest and dishonest people. Honest people rarely say deliberate lies. While we may assume people are honest, once they have said a few lies, we may assume it's more likely they aren't honest. [Recall that the upper case letters are universally quantified variables.]

```
says(P,F) <-
  thinks_true(P,F) &
  relevant(P,F) &
  honest(P) &
  truthful_h(P,F).
says(P,F) <-
  thinks_true(P,F) &
  relevant(P,F) &
  dishonest(P) &
  truthful_h(P,F).
```

```
says(P,F) <-
  honest(P) &
  untruthful_h(P,F).
says(P,F) <-
  dishonest(P) &
  untruthful_d(P,F).
```

Here $truthful_h(P, F)$ is the atomic choice that specifies the probability that P who is honest will say something F that they think is true and relevant.

The following specify the alternatives and the corresponding probabilities:

```
random([relevant(P,F):0.05,irrelevant(P,F):0.95]).
random([honest(P):0.999,dishonest(P):0.001]).
random([truthful_h(P,F):0.9999,untruthful_h(P,F):0.0001]).
random([truthful_d(P,F):0.998,untruthful_d(P,F):0.002]).
```


Whether they think something is true depends on whether they are mistaken.

```
thinks_true(P,F) <-  
  true(F) &  
  notmistaken_t(P,F).
```

```
thinks_true(P,F) <-  
  false(F) &  
  mistaken_f(P,F).  
random([mistaken_t(P,F):0.02,notmistaken_t(P,F):0.98]).  
random([mistaken_f(P,F):0.06,notmistaken_f(P,F):0.94]).
```

We haven't axiomatised what may be false.

3.3 What is true

For most things we accept that they are just true, without any deeper explanation⁴. For other things (e.g., why someone was shot or they disappeared) we look for deeper explanations.

```
true(X) <-  
  just_true(X).  
true(says(P,X)) <-  
  says(P,X).  
true(shot(P)) <-  
  shot(X,P).  
true(disappeared(X)) <-  
  left_for_no_reason(X).  
true(disappeared(X)) <-  
  disappeared_when_criminal(X) &  
  committed_crime(X).
```

```
random([disappeared_when_criminal(X):0.8, stayed_when_criminal(X):0.2]).  
random([left_for_no_reason(P):0.001,open_in_whereabouts(P):0.999]).
```

If someone is shot, we want to explain the means and opportunity as well as the motive. However not everyone who has means and opportunity and motive actually shoots. We also need to assume that they actually shot.

```
shot(X,P) <-  
  means_opportunity_to_shoot(X,P) &
```

⁴Note that the *true* predicate is only an artifact of only wanting to quantify over individuals. We reify the relations that people may say are true.

```
motive_to_shoot(X,P) &  
actually_shot(X,P).
```

```
random([actually_shot(X,P):0.01,didnt_actually_shot(X,P):0.99]).
```

Someone has the means and opportunity to shoot someone else if they are both at the same place:

```
means_opportunity_to_shoot(X,P) <-  
  at(X,L) &  
  at(P,L).
```

```
at(X,L) <-  
  true(clerk_at(X,L)).  
at(X,L) <-  
  true(wentto(X,L)).
```

The fact the someone is a vicious SOB may be a motive to shoot. But the vicious SOB only has a motive to shoot some people.

```
motive_to_shoot(X,P) <-  
  true(vicious_sob(X)) &  
  vicious_sob_shot(X,P).  
motive_to_shoot(X,P) <-  
  wanted_money(X) &  
  had_money(P).
```

```
random([vicious_sob_shot(X,P):0.2,  
        vicious_sob_didnt_shoot(X,P):0.8]).
```

The most likely explanation for a blinding flash is a picture taken. Another explanation may be that the person is shot.

```
true(observed(X,blinding_flash)) <-  
  picture_taken(X).  
true(observed(X,blinding_flash)) <-  
  true(shot(X)).
```

```
random([picture_taken(X):0.06,no_picture_taken(X):0.94]).
```

A person committed a crime if they shot someone:

```
committed_crime(X) <-  
  shot(X,P).
```

There are some things we just accept without explanation:

```

random([just_true(wentto(X,Y)):0.05,
        just_false(wentto(X,Y)):0.95]).
random([just_true(owns(X,Y)):0.02,
        just_false(owns(X,Y)):0.98]).
random([just_true(clerk_at(X,Y)):0.01,
        just_false(clerk_at(X,Y)):0.99]).

random([just_true(nickname(X,Y)):0.01,
        just_false(nickname(X,Y)):0.99]).
random([just_true(vicious_sob(X)):0.01,
        just_false(vicious_sob(X)):0.99]).

```

4 Dynamics of Belief Updating

For most of the observations, we just explain them trivially. Essentially we accept them at face value, but with the assumption that they just happened to occur.

However, if there some common causes that can account for multiple observations, then they become more likely than treating the observations as independent events. The more cohesive explanations become more likely than those that assume that the co-occurrence of activities was coincidence.

For example, the most likely explanation of *says(peter, observed(peter, blinding_flash))* is that Peter is honest and there was a picture taken.

The most likely explanation of *says(peter, says(newspaper, disappeared(harry)))* is that Peter is honest but is being untruthful here.

Similarly the most likely explanation for *says(peter, says(doctor, shot(peter)))* is that Peter is honest but is being untruthful here.

However the most likely explanation for the conjunction is that Peter is dishonest.

As the other three observations are observed, the explanation that Peter was shot by Harry becomes the most likely explanation as the simple, but a priori unlikely, event explains a number of the observations.

5 Conclusion

This paper has sketched out the sort of knowledge that we can specify in the ICL. Hopefully it will give some idea of the sort of tools we can provide.

My goal is that people will see argue about whether the axioms are appropriate, whether they are true, whether they cover all of the cases and whether the probabilities are appropriate. (Of course my axioms for Tillers' example fail on all of these

counts). I would like people to suggest better rules and alternatives. I will hope that you will find the ICL a nice framework in which to think about the problems, and hopefully you can suggest features you would like in the ICL.

My scientific hypothesis is that a causal axiomatization and hypothetical reasoning (or equivalently first-order belief networks written in rule form) is a natural way to express knowledge.

The ICL code and my axiomatization is available from my web site, but it is only a prototype and is awkward to use.

A Two Explanations in Detail

The most likely explanation for the observation of Section 3.1 is:

```
[truthful_h(peter,says(newspaper,disappeared(harry))),
relevant(peter,says(newspaper,disappeared(harry))),
notmistaken_t(peter,says(newspaper,disappeared(harry))),
truthful_h(newspaper,disappeared(harry)), honest(newspaper),
relevant(newspaper,disappeared(harry)),
notmistaken_t(newspaper,disappeared(harry)),
disappeared_when_criminal(harry),
truthful_h(peter,says(doctor,shot(peter))),
relevant(peter,says(doctor,shot(peter))),
notmistaken_t(peter,says(doctor,shot(peter))),
truthful_h(doctor,shot(peter)), honest(doctor),
relevant(doctor,shot(peter)), notmistaken_t(doctor,shot(peter)),
truthful_h(peter,observed(peter,blinding_flash)),
relevant(peter,observed(peter,blinding_flash)),
notmistaken_t(peter,observed(peter,blinding_flash)),
actually_shot(harry,peter), vicious_sob_shot(harry,peter),
truthful_h(peter,vicious_sob(harry)),
relevant(peter,vicious_sob(harry)),
notmistaken_t(peter,vicious_sob(harry)),
just_true(vicious_sob(harry)),
truthful_h(peter,clerk_at(harry,hvstore)),
relevant(peter,clerk_at(harry,hvstore)),
notmistaken_t(peter,clerk_at(harry,hvstore)),
just_true(clerk_at(harry,hvstore)),
truthful_h(peter,wentto(peter,hvstore)), honest(peter),
relevant(peter,wentto(peter,hvstore)),
notmistaken_t(peter,wentto(peter,hvstore)),
```

```
just_true(wentto(peter,hvstore))]
```

This has a prior probability of 2.6485e-019.

The second most likely explanation is:

```
[untruthful_d(peter,says(newspaper,disappeared(harry))),  
untruthful_d(peter,says(doctor,shot(peter))),  
truthful_h(peter,observed(peter,blinding_flash)),  
relevant(peter,observed(peter,blinding_flash)),  
notmistaken_t(peter,observed(peter,blinding_flash)),  
picture_taken(peter),untruthful_d(peter,vicious_sob(harry)),  
untruthful_d(peter,clerk_at(harry,hvstore)),  
truthful_h(peter,wentto(peter,hvstore)),dishonest(peter),  
relevant(peter,wentto(peter,hvstore)),  
notmistaken_t(peter,wentto(peter,hvstore)),  
just_true(wentto(peter,hvstore))]
```

This has a prior probability of 1.15225e-019.

References

- Nilsson, N. J. (1991). Logic and artificial intelligence, *Artificial Intelligence* **47**: 31–56.
- Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann, San Mateo, CA.
- Pearl, J. (1999). Reasoning with cause and effect, *Proc. 16th International Joint Conf. on Artificial Intelligence (IJCAI-99)*, pp. 1437–1449.
- Pearl, J. (2000). *Causality: Models, Reasoning and Inference*, Cambridge University Press.
- Poole, D. (1993a). Logic programming, abduction and probability: A top-down anytime algorithm for computing prior and posterior probabilities, *New Generation Computing* **11**(3–4): 377–400.
- Poole, D. (1993b). Probabilistic Horn abduction and Bayesian networks, *Artificial Intelligence* **64**(1): 81–129.
- Poole, D. (1996). Probabilistic conflicts in a search algorithm for estimating posterior probabilities in Bayesian networks, *Artificial Intelligence* **88**: 69–100.

- Poole, D. (1997). Probabilistic partial evaluation: Exploiting rule structure in probabilistic inference, *Proc. 15th International Joint Conf. on Artificial Intelligence (IJCAI-97)*, Nagoya, Japan, pp. 1284–1291.
URL: <http://www.cs.ubc.ca/spider/poole/abstracts/pro-pa.html>
- Poole, D. (2000). Abducing through negation as failure: stable models in the Independent Choice Logic, *Journal of Logic Programming* **44**(1–3): 5–35.
URL: <http://www.cs.ubc.ca/spider/poole/abstracts/abnaf.html>
- Poole, D., Mackworth, A. & Goebel, R. (1998). *Computational Intelligence: A Logical Approach*, Oxford University Press, New York.
- Savage, L. J. (1972). *The Foundation of Statistics*, 2nd edn, Dover, New York.
- Von Neumann, J. & Morgenstern, O. (1953). *Theory of Games and Economic Behavior*, third edn, Princeton University Press, Princeton, NJ.