The Independent Choice Logic: A pragmatic combination of logic and decision theory

David Poole

University of British Columbia
The independent choice logic influences

Independent Choice Logic

- Logic
- Uncertainty using decision theory
- Logic specifies consequences of choices
- Game theory
- Abduction
- Who chooses assumptions?
- Bayesian networks / influence diagrams
- Rule-structured conditional probability tables
- Logic-based state and transitions
- Dynamical systems / POMDPs
Overview

➤ Knowledge representation, logic, decision theory.
➤ Abduction + who chooses the assumptions?
➤ Logic + handle uncertainty using decision theory.
➤ Bayesian networks + rule-structured conditional probability tables.
➤ Dynamical systems and logic.
Knowledge Representation

- Problem
- Representation
- Solve
- Solution
- Interpret
- Compute
- Output
- Find compact / natural representations, exploit features of representation for computational gain.
- Approximate the solution, not the problem!
- Simplicity.
The problem: what should an agent do?

➤ It depends on its goals / background knowledge / (experience) / observations.

➤ Two normative traditions:
  ➤ **logic** semantics (symbols have meaning), proofs
  ➤ **decision / game theory** tradeoffs under uncertainty (use logic at the object-level, not the meta-level)
Assumption-based reasoning

➤ Given background knowledge / facts \( F \) and assumables / possible hypotheses \( H \),

➤ An explanation of \( g \) is a set \( D \) of assumables such that

\[
F \cup D \not\models false
\]
\[
F \cup D \models g
\]

➤ abduction is when \( g \) is given and you want \( D \)

➤ default reasoning / prediction is when \( g \) is unknown
Who chooses the assumptions?

➤ **sceptical** adversary chooses the assumptions

➤ **credulous** agent chooses what assumptions it likes

➤ **probabilistic** nature gets to choose assumptions

<table>
<thead>
<tr>
<th></th>
<th>sceptical</th>
<th>credulous</th>
<th>probabilistic</th>
</tr>
</thead>
<tbody>
<tr>
<td>prediction</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>abduction</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Independent Choice Logic

➤ C, the choice space is a set of alternatives.
An alternative is a set of atomic choices.
An atomic choice is a ground atomic formula.
An atomic choice can only appear in one alternative.

➤ F, the facts is an acyclic logic program.
No atomic choice unifies with the head of a rule.
Example: cascaded inverters

\[
\begin{align*}
\text{C} &= \{ \{\text{ok}(i_1), \text{shorted}(i_1), \text{broken}(i_1)\} \\
&\quad \{\text{ok}(i_2), \text{shorted}(i_2), \text{broken}(i_2)\} \\
&\quad \{\text{input}(\text{on}), \text{input}(\text{off})\} \} \\
\text{F} &= \{ \text{out}(i_1, \text{on}) \leftarrow \text{ok}(i_1) \land \text{input}(\text{off}), \\
&\quad \text{out}(i_1, V) \leftarrow \text{shorted}(i_1) \land \text{input}(V), \\
&\quad \text{out}(i_1, \text{off}) \leftarrow \text{broken}(i_1), \cdots \} 
\end{align*}
\]
Abductive Characterisation of ICL

- The atomic choices are assumable.
- The elements of an alternative are mutually exclusive.
- Each alternative is controlled by an agent. They get to choose the elements of the alternative.

Note that:

- The choices are independent; the facts provide no constraints on choices.
- We can do both abduction and prediction.
Nature choosing assumptions

➤ Have a probability distribution over alternatives controlled by nature.

➤ For every alternative $\chi \in C$ that is controlled by nature, there is a function:

$$P_0 : \chi \rightarrow [0, 1]$$

such that

$$1 = \sum_{\alpha \in \chi} P_0(\alpha)$$
Independent choice logic theory

**C** is a choice space

**F,** the facts, is an acyclic logic program such that no atomic choice unifies with the head of any rule.

**A** is a finite set of agents. There is a distinguished agent 0 called “nature”.

*controller* is a function from **C** → **A**. Let

\[ C_a = \{ \chi \in C : \text{controller}(\chi) = a \}. \]

**P**₀ is a function \( \bigcup \mathcal{C}_0 \to [0, 1] \) such that \( \forall \chi \in \mathcal{C}_0, \sum_{\alpha \in \chi} P_0(\alpha) = 1. \)
Probabilities of propositions

Suppose the rules are disjoint

\[ a \leftarrow b_1 \]
\[
\ldots \quad b_i \land b_j \text{ for } i \neq j \text{ can’t be true} \]
\[ a \leftarrow b_k \]

We can define:

\[ P(g) = \sum_{E \text{ is a minimal explanation of } g} P(E) \]

\[ P(E) = \prod_{h \in E} P_0(h) \]

\( P \) satisfies the axioms of probability.
Conditional Probabilities

\[ P(g|e) = \frac{P(g \land e)}{P(e)} \quad \text{← explain } g \land e \quad \text{← explain } e \]

➤ Given evidence \( e \), explain \( e \) then try to explain \( g \) from these explanations.

➤ The explanations of \( g \land e \) are the explanations of \( e \) extended to also explain \( g \).

➤ Probabilistic conditioning is abduction + prediction.
Logic for reasoning

➤ How can we reconcile the normative arguments for logic and decision theory?

➤ Logic provides:
  ➤ Symbols have denotation.
  ➤ Way to determine truth of sentences (semantics).
  ➤ Proof procedures.

... so we need at least the first order predicate calculus.
Logic and decisions

Claim: disjunction is a stupid way to handle uncertainty.

Idea: let's try to handle all uncertainty using Bayesian decision theory / game theory.

We want: the strongest logic that includes no uncertainty. Let's use acyclic logic programs (including negation as failure).

All we have lost is the ability to handle uncertainty using disjunction!
Game theory

The **strategic form of a game** [von Neumann and Morgenstern, 1953]

- Multiple agents each get to choose a strategy.
- Nature has a probability distribution over strategies.
- A complete game (choice by every agent including nature) has a utility.
- Each player chooses its strategy to maximize its utility.

We use a logic program to specify the consequences of choices.
Semantics of ICL

➤ A **total choice** is a set containing exactly one element of each alternative in $C$.

➤ For each total choice $\tau$ there is a **possible world** $w_\tau$.

➤ Formula $f$ is **true** in $w_\tau$ (written $w_\tau \models f$) if $f$ is true in the (unique) stable model of $F \cup \tau$. 
Meaningless Example

\[ C = \{\{c_1, c_2, c_3\}, \{b_1, b_2\}\} \]

\[ F = \{ f \leftarrow c_1 \land b_1, \quad f \leftarrow c_3 \land b_2, \]
\[ d \leftarrow c_1, \quad d \leftarrow \neg c_2 \land b_1, \]
\[ e \leftarrow f, \quad e \leftarrow \neg d, \]
\[ u(a_1, 5) \leftarrow \neg e, \quad u(a_1, 0) \leftarrow e \land f, \]
\[ u(a_1, 9) \leftarrow e \land \neg f, \]
\[ u(a_2, 7) \leftarrow d, \quad u(a_2, 2) \leftarrow \neg d \} \]
There are 6 possible worlds:

\[
\begin{align*}
  w_1 & \models c_1 \ b_1 \ f \ d \ e \ u(a_1, 0) \ u(a_2, 7) \\
  w_2 & \models c_2 \ b_1 \ \sim f \ \sim d \ e \ u(a_1, 9) \ u(a_2, 2) \\
  w_3 & \models c_3 \ b_1 \ \sim f \ d \ \sim e \ u(a_1, 5) \ u(a_2, 7) \\
  w_4 & \models c_1 \ b_2 \ \sim f \ d \ \sim e \ u(a_1, 5) \ u(a_2, 7) \\
  w_5 & \models c_2 \ b_2 \ \sim f \ \sim d \ e \ u(a_1, 9) \ u(a_2, 2) \\
  w_6 & \models c_3 \ b_2 \ f \ \sim d \ e \ u(a_1, 0) \ u(a_2, 2)
\end{align*}
\]
Abductive and semantic view

- The explanations of \( g \) form a concise (DNF) description of the worlds where \( g \) is true.

- The abductive characterisation is sound and complete with respect to the semantics.

- The possible worlds view shows how we can handle negation as failure and non-disjoint rules.

- The abductive characterisation can be extended to include negation as failure and non-disjoint rules.
Probabilities of Propositions

➤ When all choices are made by nature (& finite $C$):

\[
P(w_\tau) = \prod_{a \in \tau} P_0(a)
\]

\[
P(f) = \sum_{\tau : w_\tau \models f} P(w_\tau)
\]

➤ Theorem: the probabilities from the semantic view correspond to the probabilities in the abductive view.
Surely independent hypotheses aren’t powerful enough for real applications.

➤ No! Independent hypotheses can represent any probability distribution.

➤ The ICL can represent any probability represented in a Bayesian network.

➤ The ICL is more compact than a Bayesian network.
Factorization of probability distribution

➤ Bayesian networks provide a decomposition of a joint probability.

➤ Totally order the variables, $x_1, \ldots, x_n$.

\[
P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i|x_{i-1} \ldots x_1)
\]

\[
= \prod_{i=1}^{n} P(x_i|\pi_{x_i})
\]

➤ $\pi_{x_i}$ are parents of $x_i$: set of variables such that the predecessors are independent of $x_i$ given its parents.
(Bayesian) Belief Networks

➤ Graphical representation of dependence.
➤ DAGs with nodes representing random variables.
➤ Arcs from parents of a node into the node.
➤ If $b_1, \cdots, b_k$ are the parents of $a$, we have an associated conditional probability table

\[
P(a|b_1, \cdots, b_k)
\]
Bayesian Network for Overhead Projector

- power_in_building
  - light_switch_on
    - room_light_on
      - alan_reading_book
  - power_in_wire
    - projector_switch_on
      - projector_plugged_in

- power_in_projector
  - lamp_works
  - projector_lamp_on
    - ray_is_awake
      - ray_says_"screen is dark"
    - screen_lit_up
      - mirror_working
Bayesian networks as logic programs

\[
\text{projector\_lamp\_on} \leftarrow \\
\quad \text{power\_in\_projector} \land \\
\quad \text{lamp\_works} \land \\
\quad \text{projector\_working\_ok}. \quad \text{atomic choice}
\]
\[
\text{chosen by nature}
\]

\[
\text{projector\_lamp\_on} \leftarrow \\
\quad \text{power\_in\_projector} \land \\
\quad \sim \text{lamp\_works} \land \\
\quad \text{working\_with\_faulty\_lamp}.
\]
Probabilities of hypotheses

\[
P_0(\text{projector\_working\_ok})
= P(\text{projector\_lamp\_on} \mid \text{power\_in\_projector} \land \text{lamp\_works})
\]
— provided as part of Bayesian network
Mapping Bayesian networks into ICL

Translation into the rules:

\[ a(V) \leftarrow b_1(V_1) \land \cdots \land b_k(V_k) \land h(V, V_1, \ldots, V_k). \]

and the alternatives:

\[ \forall v_1 \cdots \forall v_k \{ h(v, v_1, \ldots, v_k) \mid v \in \text{domain}(a) \} \in C \]
Bayesian networks and the ICL

➢ The probabilities for the Bayesian network and the ICL translation are identical.

➢ In the translation, the ICL requires the same number of probabilities as the Bayesian network.

➢ Often the ICL theory is more compact than the corresponding conditional probability table.

➢ The probabilistic part of the ICL can be seen as a representation for the independence of Bayesian networks.
What can we learn from the mapping?

ICL adds
➤ rule-structured conditional probability tables
➤ logical variables and negation as failure in rules
➤ arbitrary computation in the network
➤ choices by other agents
➤ algorithms

Bayesian networks add
➤ theory of causation
➤ algorithms
➤ ties to MDPs, Neural networks, …
Where to now?

**Algorithms:**
- Extend model-based diagnosis algorithms to Bayes nets
- Extend Bayes net algorithms to exploit rule-structure

**Decisions:**
- Choices make by various agents
- Utility
- Actions contingent on observations (conditional plans)
Dynamical systems:

➤ Represent change using the situation calculus
➤ Logic-based partially observable Markov decision processes

Learning:

➤ represent the task of learning in the ICL
➤ combining inductive logic programming and Bayesian network (and neural network) learning
Decision Theory

➤ An agent makes choices to maximize its expected utility.

➤ What an agent should do now depends on what it will do in the future.

➤ What an agent will do in the future depends on what it will observe.

➤ An agent adopts a policy (strategy), a function from observations (and past actions) into actions.
Sequential decision problem

disease

results

test
treat

utility
Representing the decision problem

You represent the problem with rules such as:

\[
\begin{align*}
\text{result}(\text{none}) & \leftarrow \sim \text{test} \\
\text{result}(\text{positive}) & \leftarrow \text{test} \land \text{disease} \land \sim \text{false}_\text{neg}. \\
\text{result}(\text{positive}) & \leftarrow \text{test} \land \sim \text{disease} \land \text{false}_\text{pos}. \\
\text{utility}(20) & \leftarrow \text{test} \land \text{disease} \land \text{treat}.
\end{align*}
\]

A policy is something like:

\[
\begin{align*}
\text{test}.
\end{align*}
\]

\[
\begin{align*}
\text{treat} & \leftarrow \text{result}(\text{positive}).
\end{align*}
\]

All of these rules imply an expected utility.
Example: a simple robot domain
Axiomatising the simple robot domain

\[
\text{carrying}(\text{key}, T + 1) \leftarrow \\
\text{do}(\text{pickup}(\text{key}), T) \land \\
\text{at}(\text{robot}, \text{Pos}, T) \land \text{at}(\text{key}, \text{Pos}, T) \land \\
\text{pickup\_succeeds}(T).
\]

\[
\text{carrying}(\text{key}, T + 1) \leftarrow \\
\text{do}(A, T) \land \\
\text{carrying}(\text{key}, T) \land \\
A \neq \text{putdown}(\text{key}) \land A \neq \text{pickup}(\text{key}) \land \\
\text{keeps\_carrying}(\text{key}, T).
\]
Alternatives

\[ \forall T \{\text{pickup\_succeeds}(T), \text{pickup\_fails}(T)\} \in C_0 \]

\( P_0(\text{pickup\_succeeds}(T)) \) is the probability the robot is carrying the key after the \( \text{pickup}(\text{key}) \) action when it was at the same position as the key, and wasn’t carrying the key.

\[ \forall S \{\text{keeps\_carrying}(\text{key}, T), \text{drops}(\text{key}, T)\} \in C_0 \]
Imperfect Sensors

A sensor is symptomatic of what is true in the world.

\[
\text{sense}(\text{at}_\text{key}, T) \leftarrow \\
\text{at} (\text{robot}, P, T) \land \\
\text{at} (\text{key}, P, T) \land \\
\text{sensor}\_\text{true}\_\text{pos}(T).
\]

\[
\text{sense}(\text{at}_\text{key}, T) \leftarrow \\
\text{at} (\text{robot}, P_1, T) \land \\
\text{at} (\text{key}, P_2, T) \land \\
P_1 \neq P_2 \land \\
\text{sensor}\_\text{false}\_\text{pos}(T).
\]
Utility Axioms

Utility complete if $\forall w_\tau \forall T$, there exists unique $U$ such that $w_\tau \models utility(U, T)$

\[
utility(R + P, T) \leftarrow
\]
\[
prize(P, T) \land
\]
\[
resources(R, T).
\]

\[
prize(-1000, T) \leftarrow crashed(T).
\]

\[
prize(1000, T) \leftarrow in_{\_}lab(T) \land \sim crashed(T).
\]

\[
prize(0, T) \leftarrow \sim in_{\_}lab(T) \land \sim crashed(T).
\]
resources(200, s_0).

resources(R – Cost, T + 1) ←

do(goto(To, Route), T) \land

at(robot, From, T) \land

pathcost(From, To, Route, Cost) \land

resources(R, T).

resources(R – 10, T + 1) ←

do(A, T) \land

\neg gotoaction(A) \land

resources(R, T).

gotoaction(goto(Pos, T)).
Example Policy

**do**(*pickup*(key), T) ←

*sense*(at_key, T) ∧

¬*carrying*(key, T).

**do**(*goto*(door1, direct), T) ←

*carrying*(key, T).

**do**(*goto*(key_cupboard, direct), T) ←

¬*sense*(at_key, T) ∧

¬*carrying*(key, T).
Conclusions

➤ ICL is a representation that combines logic and Bayesian decision theory / game theory.

➤ Generalises acyclic logic programs, Bayesian networks, the strategic form of a game, …

➤ All rules can be interpreted logically. All numbers can be interpreted as probabilities.

➤ It’s (reasonably) simple.

➤ Applications: diagnosis, robot control, multimedia presentation, user modelling, …