Multi-agent actions under uncertainty: situation calculus, discrete time, plans and policies

David Poole

University of British Columbia
The problem and Solution

Problem: determine what an agent should do based on background knowledge, preferences and what it observes.

Basis for preferences and uncertainty: Bayesian decision theory. Alternatives: goals, satisficing.

Problem representation: independent choice logic. Alternatives: Bayesian networks, MDPs, FOPC, …

Action representation: situation calculus. Alternatives: discrete or continuous time.

Agent function represented as conditional plan. Alternative: policies.
Logic and Uncertainty

Choice:

- Rich logic including all of first-order predicate logic (e.g., Bacchus) — use both probability and disjunction to represent uncertainty.

- Weaker logic where all uncertainty is handled by Bayesian decision theory. The underlying logic contains no uncertainty — uncertainty is in terms of probabilities, decisions and utilities.
Independent Choice Logic

independent choices + acyclic logic program to give consequence of choices.

extension of pure Prolog with negation as failure; rules have their normal logical meaning.

all numbers can be consistently interpreted as probabilities.

extension of Bayesian networks: same notion of ‘causation’; can express structured probability tables, logical variables.

independent hypotheses: if there is a dependence we invent a new hypothesis to explain the dependence.
Independent choice logic

An independent choice logic theory is built from:

$C_0$ ‘nature’s choice space’ is a set of alternatives.

An alternative is a set of atomic choices.

An atomic choice is a ground atomic formula.

$F$ the facts is an acyclic logic program such that no atomic choice unifies with the head of any rule. Can include negation as failure.
A **total choice** is a set containing exactly one element of each alternative in $C_0$.

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 $F \cup \tau$. 
Independent choice logic

An independent choice logic theory can also contain:

A called the action space, is a set of primitive actions that the agent can perform.

O the observables is a set of terms.

$P_0$ is a function $\cup C_0 \rightarrow [0, 1]$.

Probability distribution over alternatives:

$\forall \chi \in C_0, \sum_{\alpha \in \chi} P_0(\alpha) = 1.$
Temporal models in ICL

ICL is independent of any model of time. E.g.,

• Time implicit: action chosen depends on history:
  \[ do(A) \leftarrow do\_choice(A, C) \land history(C) \]
  \[ \forall C \{do\_choice(A, C) : A \text{ possible action}\} \in C_\alpha \]

• Explicit time: discrete Markovian
  \[ do(A, T) \leftarrow do\_choice(A, S) \land state(S, T) \]
  \[ state(S', T + 1) \leftarrow state\_trans(S, S') \land state(S, T) \]
  \[ \forall S \{do\_choice(A, S) : A \text{ possible action}\} \in C_\alpha \]
  \[ \forall S \{state\_trans(S, S') : S' \text{ state}\} \in C_0 \]

• Situation-based time, actions specified in program.
$s_0$ is a situation and $do(A, S)$ is a situation if $A$ is an action and $S$ is a situation.

Deterministic case: the trajectory of actions by the (single) agent determines what is true — situation=state.

With uncertainty, the trajectory of an agent’s actions does not determine what is true.

Choice:
- keep the semantic conception of situation=state,
- or keep the syntactic form, so situation≠state, but situations have simple form.
In general there will be a probability distribution over states for a situation.

The agent’s actions are treated very differently from exogenous actions.
Situation Calculus in ICL

A possible world is temporally extended — specifies a truth value for every fluent in every situation.

Use standard situation calculus rules to specify what is true after an action — body of rules can include atomic choices.

Robot programs select situations in possible worlds.

Programs can be contingent on observations: the robot will observe different things in different possible worlds.

Actions have no preconditions — they can always be attempted.
Situation Calculus in ICL: Example

\begin{align*}
carrying(key, do(pickup(key), S)) & \leftarrow \\
& at(robot, Pos, S) \land \\
& at(key, Pos, S) \land \\
& pickup\_succeeds(S).
\end{align*}

\begin{align*}
carrying(key, do(A, S)) & \leftarrow \\
& carrying(key, S) \land \\
& A \neq putdown(key) \land \\
& A \neq pickup(key) \land \\
& keeps\_carrying(key, S).
\end{align*}
Alternatives

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

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

\[ \forall S \{ \text{keeps\_carrying}(key, S), \text{drops}(key, S) \} \in C_0 \]
Utility Axioms

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

$$utility(R + P, S) \leftarrow$$

$$prize(P, S) \land$$

$$resources(R, S).$$

$$prize(-1000, S) \leftarrow crashed(S).$$

$$prize(1000, S) \leftarrow in\_lab(S) \land \sim crashed(S).$$

$$prize(0, S) \leftarrow \sim in\_lab(S) \land \sim crashed(S).$$
\text{resources}(200, s_0).

\text{resources}(R - \text{Cost}, \text{do}(\text{goto}(\text{To}, \text{Route}), S)) \leftarrow \text{at}(\text{robot}, \text{From}, S) \land \text{pathcost}(\text{From}, \text{To}, \text{Route}, \text{Cost}) \land \text{resources}(R, S).

\text{resources}(R - 10, \text{do}(A, S)) \leftarrow \\
\sim \text{gotoaction}(A) \land \text{resources}(R, S).

\text{gotoaction}(\text{goto}(A, S)).
Imperfect Sensors

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

\[\text{sense(at\_key, S) } \leftarrow \]
\[\text{at(robot, P, S) } \land \]
\[\text{at(key, P, S) } \land \]
\[\text{sensor\_true\_pos}(S).\]

\[\text{sense(at\_key, S) } \leftarrow \]
\[\text{at(robot, P_1, S) } \land \]
\[\text{at(key, P_2, S) } \land \]
\[P_1 \neq P_2 \land \]
\[\text{sensor\_false\_pos}(S).\]
A **conditional plan** can use sequential composition and conditionals.

Plans select situations in worlds. The plan:

\[
\text{\textbackslash{}text{a\textbackslash{}text{}; if \text{c} then \text{b} else \text{d}; \text{e} endIf\text{}; \text{g}}
\]

selects situation \( \text{do(g, do(b, do(a, s_0)))) \) in \( w_\tau \)

if \( \text{sense(c, do(a, s_0))} \) is true in \( w_\tau \)

selects situation \( \text{do(g, do(e, do(d, do(a, s_0))))} \) in \( w_\tau \)

if \( \text{sense(c, do(a, s_0))} \) is false in \( w_\tau \).
Plans select situations in worlds

We can recursively define $do(P, W, S_1, S_2)$ which is true if doing plan $P$ in world $W$ takes us from situation $S_1$ to $S_2$.

... in pseudo Prolog:

\[
do(skip, W, S, S).
\]

\[
do(A, W, S, do(A, S)) \leftarrow
\]

\[\text{primitive}(A).\]

\[
do((P; Q), W, S_1, S_3) \leftarrow
\]

\[\text{do}(P, W, S_1, S_2) \land
\]

\[\text{do}(Q, W, S_2, S_3).\]
\[ \text{do}((\text{if } C \text{ then } P \text{ else } Q \text{ endIf}), W, S_1, S_2) \leftarrow \]
\[ W \models \text{sense}(C, S_1) \land \]
\[ \text{do}(P, W, S_1, S_2). \]
\[ \text{do}((\text{if } C \text{ then } P \text{ else } Q \text{ endIf}), W, S_1, S_2) \leftarrow \]
\[ W \not\models \text{sense}(C, S_1) \land \]
\[ \text{do}(Q, W, S_1, S_2). \]
The **expected utility** of plan $P$ is

$$
\varepsilon(P) = \sum_{\tau} p(w_{\tau}) \times u(w_{\tau}, P)
$$

where $u(W, P)$ is the utility of plan $P$ in world $W$:

$$
u(W, P) = U \text{ if } W \models \text{utility}(U, S)$$

where $do(P, W, s_0, S)$

$p(w_{\tau})$ is the probability of world $w_{\tau}$:

$$
p(w_{\tau}) = \prod_{\chi_0 \in \tau} P_0(\chi_0)$$
Other Work

Exponentially more compact than probabilistic STRIPS: E.g., each predicate $p_i$ persists stochastically and independently through a wait:

$$p_i(do(wait, S)) \leftarrow persists_{p_i}(S) \land p_i(S) \in F \text{ for each } p_i$$

$$\{persists_{p_i}(S), stops_{p_i}\} \in C_0 \text{ for each } p_i$$

Similar to action networks [Boutilier et al. 95] but doesn’t need $\#actions \times \#predicates$ space — this the frame problem!

Plans correspond to policy trees of finite stage POMDPs [Kaelbling et al. 96].

Conditional plans are like Levesque[AAAI-96]’s robot plans.
Policies

Can axiomatize change using temporal model (e.g., event calculus).

Reactive Policy:

\[ do(\text{pickup(key)}, T) \leftarrow \]
\[ \text{sense(at\_key, T)} \land \]
\[ \text{recall(want\_key, T)}. \]
Conclusion

- Combine situation calculus and Bayesian decision theory.
- Allow conditional plans and noisy sensors and effectors.
- Notion of goal expanded to utilities.
- Plans or policies have expected values.
- Planning: finding (approximately) optimal plan/policy.
- Paper explores muti-agents and reactive policies vs plans.