Decision-theoretic planning via probabilistic programming

Vaishak Belle
University of Edinburgh
Alan Turing Institute

References

- Planning in hybrid relational MDPs. Nitti, D.; Belle, V.; De Laet, T.; and De Raedt, L. Machine Learning, 1--28. 2017.
- https://dtai.cs.kuleuven.be/problog
- Probabilistic Inference in Hybrid Domains by Weighted Model Integration. Belle, V.; Passerini, A.; and Van den Broeck, G. In IJCAI, 2015.
- Hybrid Probabilistic Logic Programming. D. Nitti. PhD thesis, 2016.
- Hashing-based Approximate Probabilistic Inference in Hybrid Domains. Belle, V.; Van den Broeck, G.; and Passerini, A. In UAI, 2015.
- The Symbolic Interior Point Method. Mladenov, M.; Belle, V.; and Kersting, K. In AAAI, 2017.

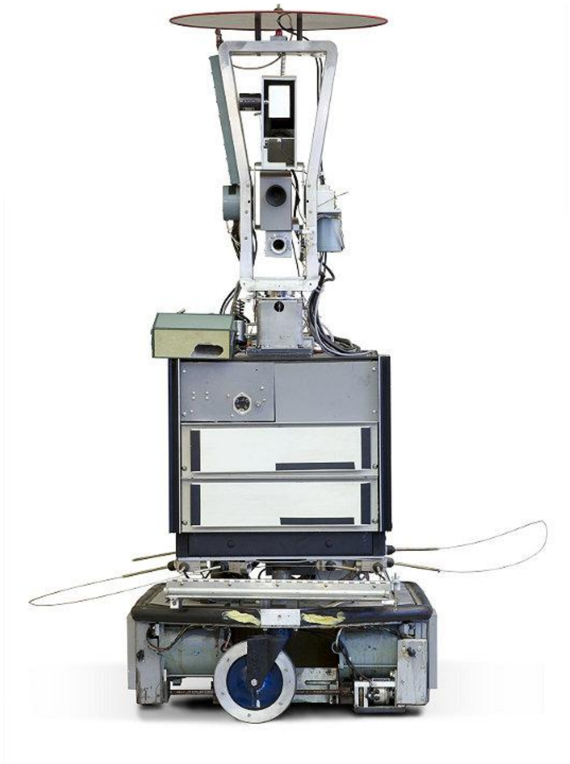
What’s on for today?

Automated planning in non-trivial stochastic domains

Transparent specification language with generic solution scheme

Computational outlook on open issues

Automated planning

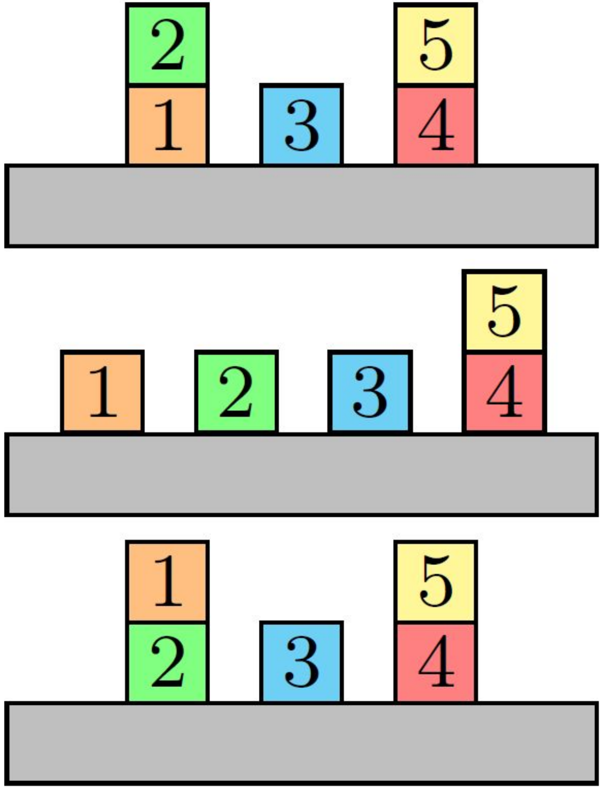


Shakey the robot [Fikes & Nilsson, 1971]

Synthesize **action sequence** to achieve goals

A world of blocks

act:	Pickup(x)
pre:	OnTable(x),Clear(x),Handempty
add:	Holding(x)
del:	OnTable(x),Clear(x),Handempty

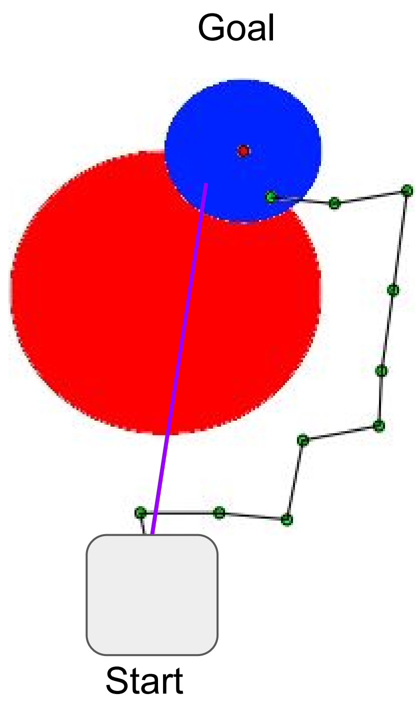


Planning in the real (noisy) world

Actions may be **stochastic**

Actions and states may be **continuous/discrete/mixed**

States may be defined over **unknown (number of) objects**



NASA Mars Rover (Bresina et al, UAI-02)

A set of initial conditions, which may involve uncertainty about continuous quantities like temperature, energy available, solar flux, and position.

A set of possible actions.

A set of certain and uncertain effects that describe the world following the action. Uncertain effects on continuous variables are characterized by probability distributions.

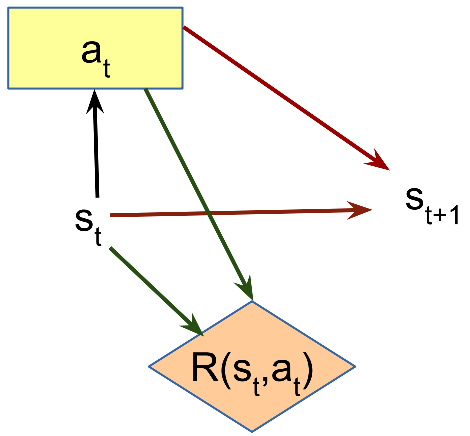
The problem that we have just described is essentially a decision-theoretic planning problem.

Markov Decision Processes

Underlying decision-theoretic framework in game theory, recommendation systems, robotics, etc.

Compute **policy:** maps states and time steps to actions

Objective: maximise expected reward over horizon *t*



Maximising expected reward

$$V_d^{\pi}(s_t) = E\left[\sum_{k=0}^{d} R(s_{t+k}, a_{t+k}) | s_t, \pi\right]$$

$$V_d^{*}(s_t) = max_a\left(R(s_t, a_t) + \gamma \int_{s_{t+1}} p(s_{t+1}|s_t, a_t) V_{d-1}^{*}(s_{t+1}) ds_{t+1}\right)$$

Additional complications

Unknown current state; estimate by noisy observation: **partially observable MDPs** that can be reduced to belief MDPs

Probabilities/rewards unknown: **reinforcement learning**

1. Elegant mathematical framework, but solving the general case notoriously hard
2. How easy to describe domain with complex relationships and discovery?

Exploit structure

Monte Carlo planners that work in arbitrary MDPs are very slow in practice

Why? Only access sample traces, but do not exploit:

- Probabilities of transitions
- Structure of the planning model

States, actions more than abstract entities: instantiated over structured domain theories that express relationships and dependencies

Desiderata

A rich modelling language that allows transparent domain axiomatisation in the presence of unknowns and stochasticity

Solution scheme that leverages inherent structure

Probabilistic programming

Languages to model structured probability distributions

Make machine learning modular and enable descriptive clarity

Programming languages with stochastic primitive

Many proposed to date: Church, BLOG, Anglican, ProbLog, IBAL, etc.

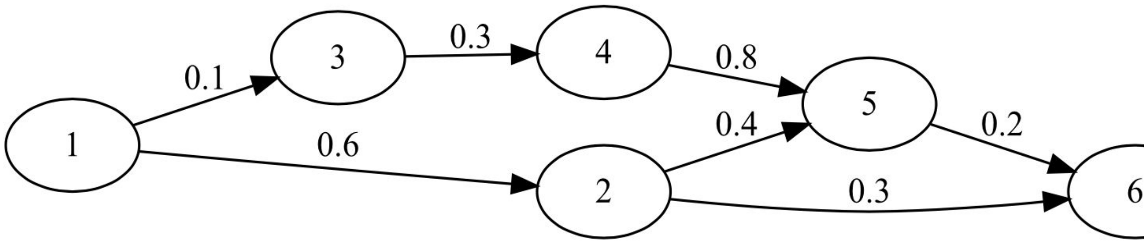
ProbLog

```
% Probabilistic facts:
0.5::heads1.
0.6::heads2.

% Rules:
twoHeads :- heads1, heads2.

% Queries:
query(heads1).
query(heads2).
query(twoHeads).
```

Two coin tosses in a sequence




```
0.6::edge(1,2).
0.1::edge(1,3).
0.4::edge(2,5).
0.3::edge(2,6).
0.3::edge(3,4).
0.8::edge(4,5).
0.2::edge(5,6).

path(X,Y) :- edge(X,Y).
path(X,Y) :- edge(X,Z),
             Y \== Z,
         path(Z,Y).

query(path(1,5)).
```

Knowledge graphs

```
father(bart, stijn).
father(bart, pieter).
father(luc, soetkin).

mother(katleen, stijn).
mother(katleen, pieter).
mother(lieve, soetkin).

parent(bart,stijn).
parent(bart,pieter).
parent(luc,soetkin).

female(alice).
female(an).
female(esther).

male(bart).
male(etienne).
male(leon).

grandmother(esther,soetkin).
grandmother(esther,stijn).
grandmother(esther,pieter).
...
```

Learning a relation

```
Candidates for iteration 4:
==================================
grandmother(A,B) :- parent(C,B), parent(A,C), parent(D,A) 0.503462603878
grandmother(A,B) :- parent(C,B), parent(A,C), parent(B,D) 0.457063711911
grandmother(A,B) :- parent(C,B), parent(A,C), male(C) 0.432528690146
grandmother(A,B) :- parent(C,B), parent(A,C), male(B) 0.432528690146
==================================
RULE LEARNED: grandmother(A,B) :- parent(C,B), parent(A,C), \+male(A) 1.0
```

Learning a relation continued

Probabilistic model

Q1: In a group of 10 people, 60 percent have brown eyes. Two people are to be selected at random from the group. What is the probability that neither person selected will have brown eyes?

Conditioning (observation)

Query

From natural language (IJCAI-17)

Unknowns, continuous distributions and dynamics

Unknown color

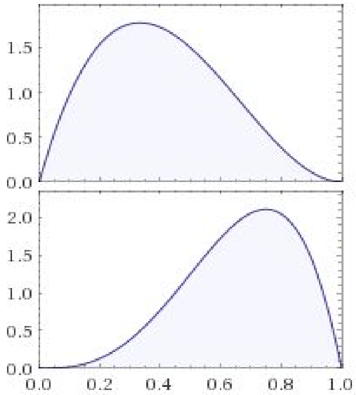
$$\overbrace{\mathtt{color(X)}}^{\text{Name random variable}} \sim \overbrace{\mathtt{uniform([black, brown])}}^{\text{distribution}} \leftarrow \overbrace{\mathtt{material(X)} \sim= \mathtt{wood}.}^{\text{conditions (body)}}$$

Unknown physical size

$$\mathtt{material(X)} \sim \mathtt{finite([0.3:wood, 0.7:metal])} \leftarrow \mathtt{between(1, N, X)}.$$

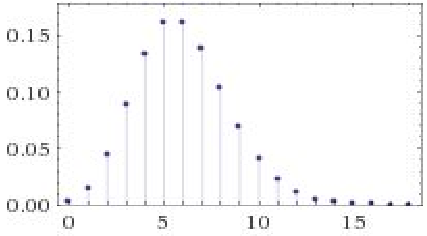
$$\mathtt{size(X)} \sim \mathtt{beta(2, 3)} \leftarrow \mathtt{material(X)} \sim= \mathtt{metal}.$$

$$\mathtt{size(X)} \sim \mathtt{beta(4, 2)} \leftarrow \mathtt{material(X)} \sim= \mathtt{wood}.$$



Unknown numbers

$\boxed{\mathtt{n} \sim \mathtt{poisson}(6).}$



(Infinite valued discrete distribution)

$$\mathtt{material}(\mathtt{X}) \sim \mathtt{finite}([0.3:\mathtt{wood}, 0.7:\mathtt{metal}]) \leftarrow \mathtt{n} \sim= \mathtt{N}, \mathtt{between}(1, \mathtt{N}, \mathtt{X}).$$

Continuous distributions, dynamics

$$\mathtt{pos_{t+1}(ID)_x} \sim \mathtt{gaussian}(\simeq(\mathtt{pos_t(ID)_x}), \sigma^2) \leftarrow$$
$$\simeq(\mathtt{move_t}) = \mathtt{ID}.$$

$$\mathtt{obsPos_{t+1}(ID)} \sim \mathtt{gaussian}(\simeq(\mathtt{pos_{t+1}(ID)}), \mathit{cov}).$$

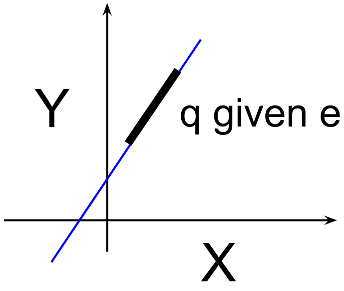
Clauses in action (object tracking)

Key inference ideas

Relevant variables: SLD resolution

Informed search: importance sampling

Avoid invalid regions: constraint propagation



$\Pr(q \mid Y = 3X+1)$

From inference to planning

Specifying MDPs

State transition model: $\mathrm{Var}_{t+1} \sim \mathrm{Distribution} \leftarrow \mathrm{Conditions}_t$

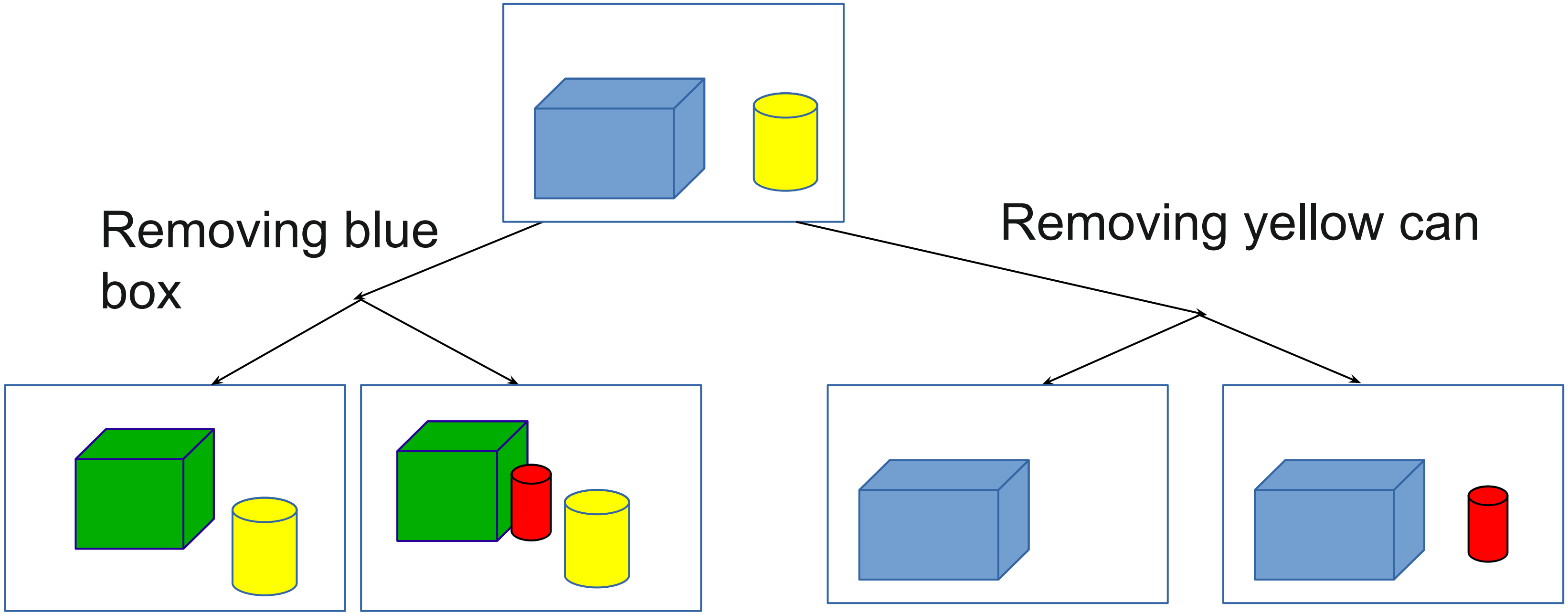
Applicable actions: $\mathrm{applicable(Action)}_t \leftarrow \mathrm{Conditions}_t$

Reward: $\mathrm{reward(R)}_t \leftarrow \mathrm{Conditions}_t$

Terminal state: $\mathrm{stop}_t \leftarrow \mathrm{Conditions}_t$

$$\mathtt{stop}_t \leftarrow \simeq(\mathtt{type}(X)_t) = \mathtt{can}.$$
$$\mathtt{reward}(20)_t \leftarrow \mathtt{stop}_t.$$
$$\mathtt{reward}(-1)_t \leftarrow \mathtt{not}(\mathtt{stop}_t).$$

Can be over unknowns (e.g., find red can)

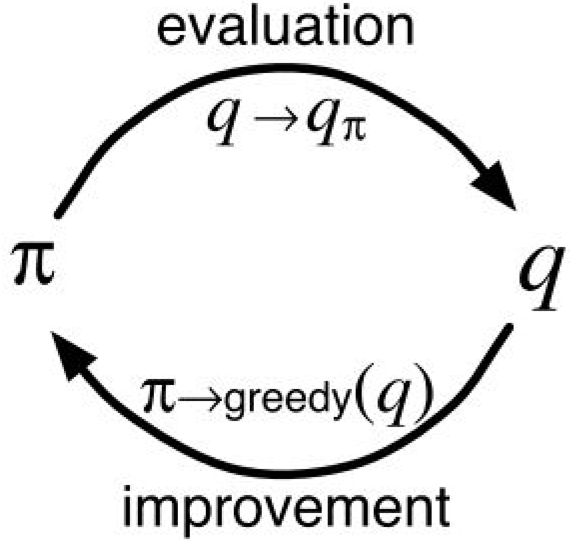


An additional function

$$V_d^{\pi}(s_t) = E\left[\sum_{k=0}^{d} R(s_{t+k}, a_{t+k}) | s_t, \pi\right]$$

$$Q_d^{\pi}(s_t, a_t) = E\left[\sum_{k=0}^{d} R(s_{t+k}, a_{t+k}) | s_t, a_t, \pi\right]$$

Computing an optimal policy



$\pi(s) \quad \leftarrow argmax_a Q^{\pi}(s, a)$

HYPE = Hybrid episodic planner

for each applicable action a in s_t^m **do**

$$\tilde{Q}_d^m(s_t^m, a) \leftarrow R(s_t^m, a) + \frac{\sum_{i=0}^{m-1} w^i \tilde{V}_{d-1}^i(s_{t+1}^i)}{\sum_{i=0}^{m-1} w^i}$$

end for

Evaluate Q-function

$$a_t^m \leftarrow policy(\{\tilde{Q}_d^m(s_t^m, a)\})$$

Select action (e.g., ε-greedy)

sample $s_{t+1}^m \sim p(s_{t+1} \mid s_t^m, a_t^m)$

Sample next state

$$G_d^m \leftarrow R(s_t^m, a_t^m) + \text{SAMPLEEPISODE}\ (d-1, s_{t+1}^m, m)$$

Recursive call

$\tilde{V}_d^m(s_t^m) \leftarrow G_d^m$

store $(s_t^m, \tilde{V}_d^m(s_t^m), d)$

return $\tilde{V}_d^m(s_t^m)$

Storing total reward or V-function value

end function

Evaluations

Domain	Planner	d	Param.	Reward	Time (s)
game1	HYPE	5	M = 1200	0.87 ± 0.11	662
	SST	5	C = 1	0.34 ± 0.15	986
	HYPE	4	M = 1200	$\mathbf{0.89 \pm 0.07}$	312
	SST	4	C = 2	0.79 ± 0.08	1538
game2	HYPE	5	M = 1200	0.67 ± 0.18	836
	SST	5	C = 1	0.14 ± 0.20	1000
	HYPE	4	M = 1200	$\mathbf{0.76 \pm 0.19}$	582
	SST	4	C = 2	0.27 ± 0.22	1528
sysadmin1	HYPE	5	M = 1200	0.94 ± 0.07	422
	SST	5	C = 1	0.47 ± 0.13	1068
	HYPE	4	M = 1200	$\mathbf{0.98 \pm 0.06}$	346
	SST	4	C = 2	0.66 ± 0.08	1527
sysadmin2	HYPE	5	M = 1200	$\mathbf{0.87 \pm 0.11}$	475
	SST	5	C = 1	0.31 ± 0.12	1062
	HYPE	4	M = 1200	0.86 ± 0.11	392
	SST	4	C = 2	0.46 ± 0.12	1532

Evaluations (2)

simplerover2	HYPE	8	M = 200	11.8 ± 0.2	38
	SST	8	C = 1	11.4 ± 0.3	48
	HYPE	9	M = 500	11.7 ± 0.2	195
	SST	9	C = 1	11.3 ± 0.3	238
	HYPE	10	M = 500	$\mathbf{11.9 \pm 0.3}$	218
	SST	10	C = 1	11.2 ± 0.3	1043
marsrover	HYPE	6	M = 6000	249.8 ± 33.5	985
	SST	6	C = 1	227.7 ± 27.3	787
	HYPE	7	M = 6000	269.0 ± 29.4	983
	SST	7	C = 1	N/A	Timeout
	HYPE	10	M = 4000	$\mathbf{296.3 \pm 19.5}$	1499
	SST	≥ 8	C = 1	N/A	Timeout

Cf. paper on results with relational abstraction

Outlook: open issues

Difficulty handling low probability observations

Guessing “good” proposal distributions hard

Bounds on computed values? (E.g., Safety-critical applications)

SAT and #SAT

Given a CNF formula,

- SAT: find a satisfying assignment
- #SAT: count satisfying assignments

$$(x \vee y) \wedge (y \vee \neg z)$$

- 5 models: (0,1,0), (0,1,1), (1,1,0), (1,1,1), (1,0,0)
- Equivalently: satisfying probability = $5/2^3$

Weighted #SAT

Polytime reduction from exact inference in discrete graphical models to weighted #SAT

Think of (1,0,0) as sequence of one heads, two tails

Exact algorithms with strong runtime bounds

Approximate algorithms with strong certificates

ProbLog reduces inference computation to Weighted #SAT

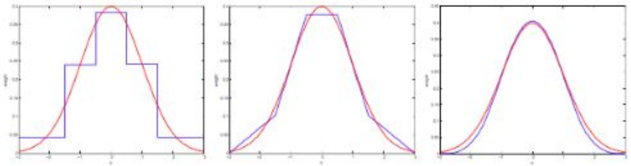
Weighted #SMT (IJCAI-15)

Constraint propagation capabilities

Exact and approximate methods have been identified

Dealing with countably infinite values (UAI-17)

Use these methods to provide tight correctness characterisations?



(a) constants (b) degree 1 (c) degree 3

$$-3 \le u \le 3$$

$$\begin{array}{rl} (2+u)^3/6 & -2 < u \le -1 \\ (4-6u^2-3u^3)/6 & -1 < u \le 0 \\ (4-6u^2+3u^3)/6 & 0 < u \le 1 \\ (2-u)^3/6 & 1 < u < 2 \end{array}$$

Summary

HYPE works in a wide range of domains: discrete, continuous, hybrid, growing vs shrinking state spaces

Systematically handles discovery of unknown objects

Exploits the probabilistic model and relational structure to provide fast solutions

Enables transparency and modularity of intricate stochastic specifications (e.g., MDP part of larger pipeline)