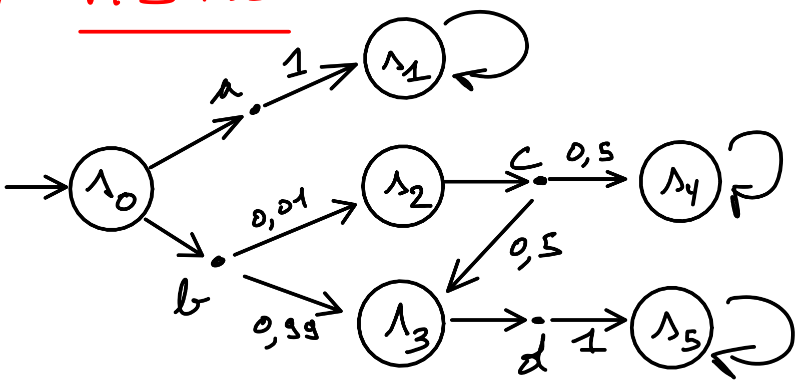
Based on [BCCF+14]: Verification of Markov decision processes using learning algorithms, ATVA 2014.

Outline

1. MDPs
2. Reachability problem
3. Classical algorithms:
   a) LP
   b) Value iterat° & Strategy iterat°
4. Using learning: bounded real-time dynamic programming (BRTDP)
5. Conclusion

I/ MDPs



MDP $\mathcal{M} = (S, A, \delta)$

- S: finite set of states
- A: " " " actions
- δ: $S \times A \to \mathcal{D}(S)$ partial probabilistic transit° funct°

Strategy $\sigma: (S \times A)^* S \to \mathcal{D}(A)$ s.t. $\delta(s, \alpha)$ is defined for each $\alpha \in \mathrm{Supp}(\sigma(\rho))$

$\sigma \in \Sigma$

Rmq Strategies may use memory and randomness.

Given an MDP $\mathcal{M}$, a state s, a strategy σ, we obtain a Markov chain (MC), hence a probability measure over paths: $\mathbb{P}^{\sigma}_{\mathcal{M}, s}$.

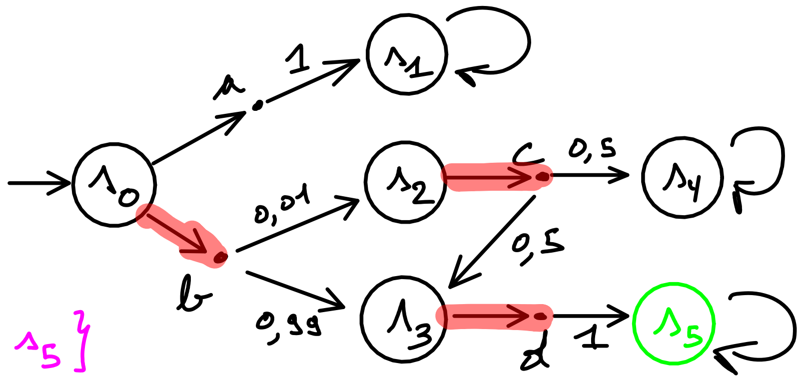
II | Reachability

The goal is to reach $T \subseteq S$ and we want a strategy σ^* that maximizes the probability to do so.

Rmk Lots of problems on MDPs can be reduced to reachability problems!

Example

$\Diamond s_5 \equiv \{ \text{paths } \pi \mid \pi \text{ reaches } s_5 \}$



$$\sup_{\sigma} \mathbb{P}^{\sigma}_{s_0}[\Diamond s_5] = 1 - 0{,}01 \cdot 0{,}5 = 0{,}995$$

Thm
Pure memoryless strategies suffice for reachability.

$\hookrightarrow$ We may replace $\sup_{\sigma \in \Sigma}$ by $\max_{\sigma \in \Sigma^{PM}}$!

$\Longrightarrow$ Cornerstone of the following algorithms.

III | Classical algorithms

a) Linear Program

Vector $(x_s)_{s \in S}$ with $x_s = \max_{\sigma} \mathbb{P}_s^{\sigma}[\Diamond T]$ is the unique solut° of the LP:

- If $s \in T$, then $x_s = 1$.
- If $\underbrace{s \not\models \exists \Diamond T}_{s \text{ not connected to } T}$, then $x_s = 0$.
- Else, $0 \leq x_s \leq 1$
  and for all $\alpha \in A(s)$ (act° enabled in s)
  $$x_s \geqslant \sum_{t \in S} \delta(s, \alpha, t) \cdot x_t \quad (*)$$

where $\sum_{s \in S} x_s$ is minimal. $(**)$

Intuit°

$(*)$ Otherwise we could increase x_s by changing α

$(**)$ We need the smaller fixed point
($x_s = 1$ is always a solut° but not the one we want)

$\Longrightarrow$ Problem $\in P$. But in practice, LP does not scale well.

b) Value & strategy iterat°

VI: approximation technique for values x_s

1. Backward reachability to determine
$\{s \mid s \models \exists\Diamond T\} \overset{!}{=} \{s \mid x_s > 0\} = Pre^*(T)$

2. For $s \in Pre^*(T) \setminus T$:
$$x_s = \lim_{n \to \infty} x_s^{(n)}$$

When $=$, Bellman equat°

where $x_s^{(0)} = 0$
and $x_s^{(n+1)} = \max \left\{ \sum_{t \in S} \delta(s,\alpha,t) \cdot x_t^{(n)} \mid \alpha \in A(s) \right\}$

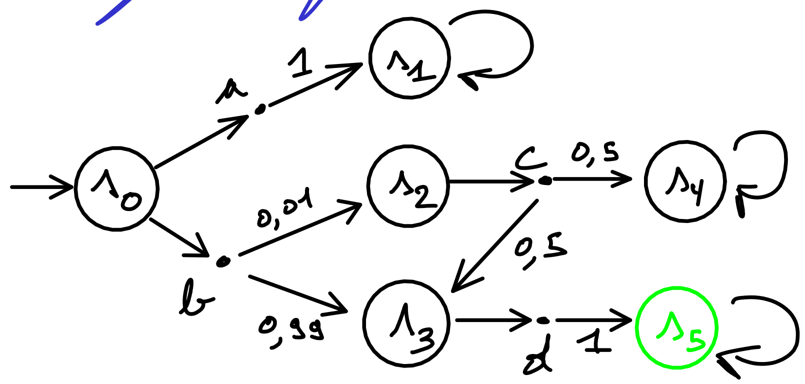
Stopping criterion:
(i) Naïve: stop when $\max_{s \in S} |x_s^{(n+1)} - x_s^{(n)}| < \varepsilon$ for some ε.
$\leftrightarrow$ Fails in some cases
(ii) Also compute an upper bound and evaluate the difference.

Thm
VI converges monotonically.

$\longrightarrow$ Exponential in the worst-case but the most efficient technique in practice.

Rmk VI does not give an (ε-)optimal scheduler directly and obtaining it from the values may be costly.

Example of VI



Iteration	s_0	s_1	s_2	s_3	s_4	s_5
0	0	0	0	0	0	1
1	0	0	0	1	0	1
2	0,99	0	0,5	1	0	1
3	0,995	0	0,5	1	0	1

$\longrightarrow$ Here we reach the limit in a finite time because no loop $\longrightarrow$ not true in general.

SI : we want to compute actual strategies, not merely values

We start as before with $x_s^{(0)}$ for $s \in Pre^*(T) \setminus T$.
Then we loop:

(i) $\sigma^{(n+1)}(s) = \arg\max_{\alpha \in A(s)} \left\{ \sum_{t \in S} \delta(s,\alpha,t) \cdot x_t^{(n)} \right\}$

(ii) Evaluate $x_s^{(n+1)} = \mathbb{P}_s^{\sigma^{(n+1)}}[\Diamond T]$ either using VI (approx.) or linear equat° system (exact but slower).

Stopping criterion : when $\sigma^{(n+1)} = \sigma^{(n)}$

Thm

SI converges monotonically.

→ Also exponential in the worst-case. Slower than VI but more precise.

Comparison

Similar loop for both VI and SI

1) Choose best actions based on current approx.

2) Update current approx.

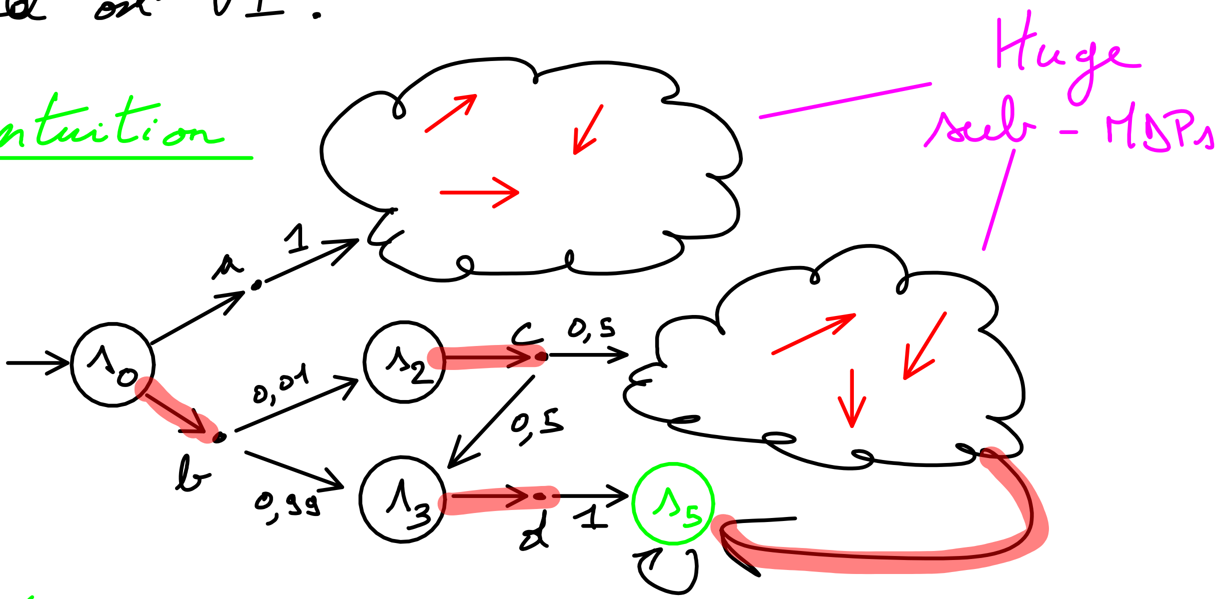
↳ In VI, update over one step using previous approx

↳ In SI, "exact" computat° based on the chosen actions.

IV | Using learning: BRTDP approach

Many different variants: I consider one based on VI.

Intuition



We do not care at all about the first cloud, and we can estimate the optimal probability up to $\varepsilon = 0{,}005$ without looking at the second cloud.

$\Longrightarrow$ We want to identify which parts of the MDP are really important and focus on them.

$\Longrightarrow$ Will speed up the search for good strategies but also permit to highlight meaningful actions, hence granting simple strategies

E.g., here $\sigma(s) = \begin{cases} b & \text{if possible} \\ \text{random} & \text{otherwise} \end{cases}$

already yields $\mathbb{P}^{\sigma}[\Diamond T] \geq 0{,}99$

$\Longrightarrow$ Related research: strategies as decision trees.

End-component (EC): strongly-connected sub-MDP (i.e., no probability leak).

$\sim$ similar to BSCCs for MCs.

Simpler case: - trivial ECs ☺ & ☹

target sink

- initial state $\bar{s}$

We want $\max_{\sigma \in \Sigma^{PM}} \mathbb{P}^{\sigma}_{\mathcal{M},\bar{s}}[\Diamond ☺] = x_{\bar{s}}$

Rmq: we will approximate $x_{\bar{s},\alpha}$
$= \sum_{t \in S} \delta(\bar{s},\alpha,t) \cdot x_t$

but $x_{\bar{s}} = \max_{\alpha \in A(s)} x_{\bar{s},\alpha}$

Intuition: we will approximate $x_{\bar{s},\alpha}$ using upper and lower bounds

$\rightsquigarrow U(\bar{s},\alpha) \longrightarrow L(\bar{s},\alpha)$

Algo

$U(\cdot,\cdot) \leftarrow 1,\ L(\cdot,\cdot) \leftarrow 0$
$L(☺,\cdot) \leftarrow 1,\ U(☹,\cdot) \leftarrow 0$ } Trivial bounds

REPEAT
$\rho \leftarrow \bar{s}$
REPEAT
$\alpha \leftarrow$ sampled uniformly from $\underset{\alpha \in A(last(\rho))}{\arg\max}\ U(last(\rho), \alpha)$

last state of prefix ρ

↔ random action maximizing U
⟹ Focus on "good" strategies

$s \leftarrow$ sampled according to $\delta(last(\rho), \alpha)$
$\rho \leftarrow \rho . \alpha . s$
UNTIL $s \in \{☺, ☹\}$ → Path ρ ends in one of the ECs

Explore phase

REPEAT
$s' \leftarrow pop(\rho)$
$\alpha \leftarrow pop(\rho)$
$s \leftarrow last(\rho)$
UPDATE(s, α)
UNTIL $\rho = \bar{s}$

} For each visited transit° update U and L (backwards)

Update phase

UNTIL $U(\bar{s}) - L(\bar{s}) < \varepsilon$

↔ Approx. is close enough.

with

$$U(s) = \max_{\alpha \in A(s)} U(s,\alpha)$$
$$L(s) = \max_{\alpha \in A(s)} L(s,\alpha)$$

UPDATE (s, α)

$$U(s,\alpha) = \sum_{s' \in S} \delta(s,\alpha,s') \cdot U(s')$$

$$L(s,\alpha) = \sum_{s' \in S} \delta(s,\alpha,s') \cdot L(s')$$

Rmq 1 This works if the MDP is known.

If δ is not known but can be sampled, UPDATE needs to be adapted

$\Longrightarrow$ It gives a PAC algorithm.

probably approximately correct

Rmq 2 The general case (non-trivial ECs) is more complex.

Intuition:

- identify ECs from long enough simulations
- contract them on the fly

V Conclusion

1) Reachability in MDPs is a key problem.
2) Textbook solut°: LP
   $\longrightarrow$ not efficient in practice
3) Classical techniques: value/strategy iterat°
   $\longrightarrow$ works well in practice
4) Mix with learning
   $\longrightarrow$ several variants (heuristics)
   $\longrightarrow$ tremendous gains: state space $\times 10^{-3}$, time $\times 10^{-2}$
5) Quantitative extensions
   $\longrightarrow$ e.g., mean-payoff