Sequential Decision-Making under uncertainty

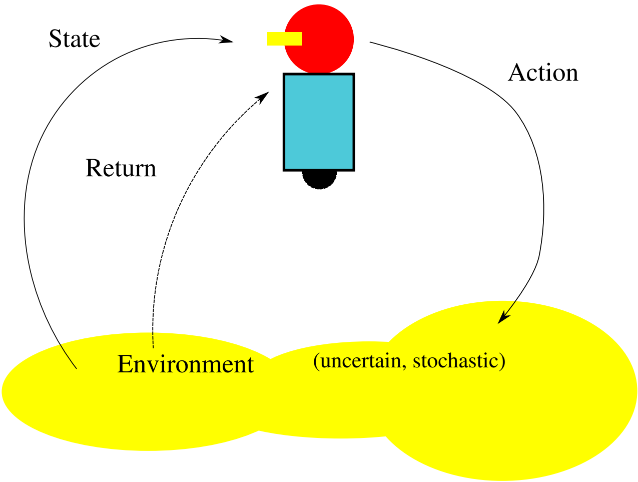
Philippe Preux

Scool

CDP DePERU, Jan. 20th, 2025.

Sequential Decision-Making under Uncertainty

What's this all about?



Formal framework: Markov decision problems $(\mathcal{E}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \zeta)$ with:

- $\mathcal{E}$: set of states,
- $\mathcal{A}$: set of actions,
- $\mathcal{P}$: transition function $\mathcal{P}(e, a, e') = Pr[e_{t+1} = e' | e_t = e, a_t = a]$,
- $\mathcal{R}$: reward function $\mathcal{R}(e, a, e')$,
- ζ : objective function.

Problem : find a policy that optimizes ζ.

Sequential Decision-Making under Uncertainty

What's this all about?

Formal framework: Markov decision problems $(\mathcal{E}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \zeta)$ with:

- $\mathcal{E}$: set of states,
- $\mathcal{A}$: set of actions,
- $\mathcal{P}$: transition function $\mathcal{P}(e, a, e') = Pr[e_{t+1} = e' | e_t = e, a_t = a]$,
- $\mathcal{R}$: reward function $\mathcal{R}(e, a, e')$,
- ζ : objective function.

Problem: find a policy that optimizes ζ.

Usual assumptions:

- Markov system, constant along time.

Remarks:

- $\mathcal{E}$ and $\mathcal{A}$ may be discrete or continuous.
- Time is usually discrete, but continuous time is also studied.

Sequential Decision-Making under Uncertainty

Wat do we mean?

Solving an MDP = solving a "reinforcement learning" problem.

- usually, $\zeta = \sum \gamma^t r_t, \gamma \in [0, 1[$.
- Then, the solution (optimal policy) depends only on the current state and it is deterministic: $\pi : \mathcal{E} \rightarrow \mathcal{A}$.
- How to find π? 3 approaches:
  - estimation of the value function $\rightsquigarrow$ policy,
  - direct policy search,
  - combination of both: actor-critic.

All 3 relies on interacting with the environment

$\rightsquigarrow$ samples of transition and reward functions

$\rightsquigarrow$ incremental estimation of V or improvement of π, or both.

Reinforcement learning

- rooted in the analysis of behavior:
  "temporal difference" method ([Sutton *et al.*; Watkins, 198x]) :
  the agent "learns when it is surprised" by the consequences of its action.

Reinforcement learning

- rooted in the analysis of behavior:
  "temporal difference" method ([Sutton *et al.*; Watkins, 198x]) :
  the agent "learns when it is surprised" by the consequences of its action.
- Algorithms based on a balanced combination of exploration and exploitation.

Reinforcement learning

- rooted in the analysis of behavior:
  "temporal difference" method ([Sutton *et al.*; Watkins, 198x]) :
  the agent "learns when it is surprised" by the consequences of its action.
- Algorithms based on a balanced combination of exploration and exploitation.
- Exact representation vs. approximate representation of V and π.
  Neural RL where a neural network represents V or π.

Reinforcement learning

- rooted in the analysis of behavior:
  "temporal difference" method ([Sutton *et al.*; Watkins, 198x]) :
  the agent "learns when it is surprised" by the consequences of its action.
- Algorithms based on a balanced combination of exploration and exploitation.
- Exact representation vs. approximate representation of V and π.
  Neural RL where a neural network represents V or π.
- Almost no theoretical result is useful in practice.

Some variants

- Observations instead of states.
  Theory: much more difficult problem.
  Practice: many applications tolerate some discrepancy between the observation and the state.
- robust RL

One-armed bandit problems

Degenerate RL problem: single state.

- Finite number of arms:
  each arm is characterized by a law describing its immediate return.
  This law is unknown.
  - performance measured by the regret.
  - usual objectives: cumulated regret minimization vs. best arm identification.

  Prototypical problem to study the exploration/exploitation trade-off.

One-armed bandit problems

- Theoretical results bound the regret of algorithms.
  E.g., smallest regret scales in $\log(\text{number-of-pulls})$ under some conditions.
- Many laws have been studied, parametric and non parametric ones.
- $+$ extension to robust objectives.
- $+$ extension to finite and infinite spaces of arms located in a metric space.

- Theory is really useful: theory guides the design and use of algorithms.
- $\rightsquigarrow$ lots of application on the web (advertizing, recommandation systems).

Some of the big questions under investigation in Scool

- little amount of data/interactions
- robustness
- applications in health and agriculture to recommand practices.

Big question for DePERU

- can we define these problems under radical uncertainty?
- If so, can we derive anything interesting?

Thank you for your attention.

Questions?