Exploration in Reinforcement Learning

Adam Jedlička, Tatiana Valentine Guy

Department of Adaptive Systems, Institute of Information Theory and Automation,
Czech Academy of Sciences, jedlicka@utia.cas.cz

In Reinforcement Learning (RL), exploration involves efficiently discovering new states, while balancing this with exploiting known states to maximize immediate rewards. This exploration-exploitation (E2E) dilemma is complex and often specific to the task and domain. The poster:

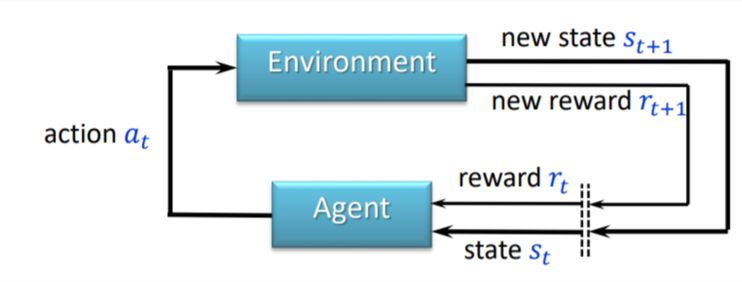
- recalls Markov Decision Process (MDP) and RL approach
- Indicates practical usefulness of RL
- briefly introduces a subsection of related exploration methods

Main Model: Markov Decision Process (MDP)

An MDP formulation of the decision-making task consists of a tuplet (S, A, R, p, γ):

- S is a set of available states $s \in S$.
- A is a set of available actions $a \in A$.
- $R : (S \times A) \rightarrow R$ is a reward function that assigns a state-action pair an immediate reward r_t (reinforcement).
- $p(s_{t+1}|s_t, a_t)$ is a transition function, i.e. the probability of moving from s_t to s_{t+1} via action a_t.
- γ is a discount factor that determines the importance of future rewards by reducing their present value and balancing immediate and long-term gains.

The basic scheme of agent´s interactions is shown in the Figure below



Solution: Reinforcement Learning (RL)

RL solves an MDP whenever the reward or transition function is unknown. It mimics human "trial-and-error" learning. The solution maximizes the cumulative reward and is based on the Bellman equation. $Q^{\pi}(s, a)$ is a Q-function expressing the utility of the state-action pair when executing decision policy π.
The basic update in the Q-learning algorithm is as follows.

Q-leaning update
input - initial Q-function Q^{old}, reward function R, learning rate α, factor γ, terminal state s_{term}, initial state s_0 and max_iter as maximum number of iterations of t.
set time $t = 0$
while $t <$ max_iter
  if $s_t = s_{term}$
    set s_t to s_0
  end if
  choose action a_t, get s_{t+1} (using on exploration algorithm).
  $Q^{new}(s_t, a_t) = Q^{old}(s_t, a_t) + \alpha(r_t + \gamma \max_{a \in A} Q^{old}(s_{t+1}, a) - Q^{old}(s_t, a_t))$
  $Q^{old}(s_t, a_t) = Q^{new}(s_t, a_t)$
  $t = t + 1$
end while
Note: when dealing with high-dimensional state spaces the traditional Q-learning would be infeasible due to the large size of the state-action space. To overcome that, deep Q-learning, which uses neural networks to approximate the Q-value function, is used.

Example of application to biology-related tasks

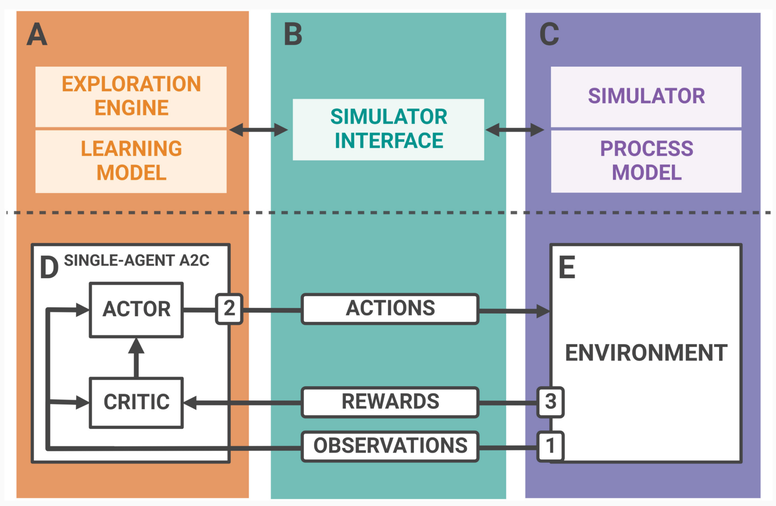
A paper [1] uses an RL-based methodology to generate optimal in-silico protocols for the simulated fabrication of epithelial sheets. In other words, a systematic method to find the best computer simulations for creating virtual models of epithelial tissues of certain sizes and shapes is proposed. The environmental model (Palacell2D, [1]) deterministically stimulates cell proliferation and apoptosis based on internal cellular pressure. The goal is twofold:

- to maximize cell growth by systematically changing the direction and volume of external pressure,
- To maximize the fraction of resulting cells within a predefined area.

This task is modeled via MDP as follows:

- **state** - number of cells and position of vertices in the PalaCell2D model
- **action** - change in direction and volume of external pressure
- **reward**- (increment in the number of cells; the fraction of cells in the target)
- **transition function** - an action defines a new state deterministically (no uncertainty is present).
- **parameter** γ - is discounting factor, $\in$ [0.95, 0.99]

The following Figure, [1], describes the structure of the solution [1]. It uses an Actor-Critic algorithm, a modification of RL that combines policy optimization (actor) and value function estimation (critic). An important part of the solution forms exploration.



Examples of other biological problems that can be solved by RL methods:

- Modeling and controlling Gene Regulatory Networks, [2].
- Developing new drugs and treatments with molecular sequences, [2].
- Protein structure prediction, [3].

Selected exploration methods

The exploration is an important task in RL that solves the E2E dilemma, which:

- Enables the agent to efficiently discover new “state-action” pairs.
- Improves the agent’s ability to adapt to new, unseen, situations.
- Helps the agent to generalize knowledge acquired.
- Avoid local optima and improves policy performance.

The algorithms below are proven to solve this dilemma and some of them can be of use for applications of RL in a wide variety of tasks including biology-related tasks.

ϵ-greedy exploration
The most basic exploration algorithm characterized by parameter ϵ, [4]

- At each time step it takes a parameter $\epsilon_t \in [0, 1]$ and generate a random number c in the same range
- If $c \leq \epsilon$ random action is chosen (exploration), otherwise, optimal action is chosen (exploitation). For Q-learning optimal action is $a_t^{opt} = argmax_{a \in A} Q(s_t, a)$
- Decrease epsilon (start from 1 and decrease to 0 over time)

Main features:

- simple to implement and understand and relatively easy tuning of ϵ
- works well for simple environments but inefficient for complex environments
- may lead to a suboptimal solution.

Boltzmann exploration
Actions are selected probabilistically based on Boltzmann distribution [4].

$$p(a_t|s_t) = \frac{e^{\frac{Q(s_t,a_t)}{\lambda_t}}}{\sum_{a_t \in \mathcal{A}} e^{\frac{Q(s_t,a_t)}{\lambda_t}}}, \quad \text{Temperature parameter } \lambda_t > 0 \text{ controls exploration.} \tag{1}$$

High temperature stimulates exploration. Temperature can be lowered over time to shift from exploration to exploitation. The main features are:

- adjusting temperature ensures a smooth transition from exploration to exploitation
- high chance to find a good action even when exploring.
- sensitivity to temperature selection and high complexity.

Upper Confidence Bound exploration

- counts the number k of times a certain action has been taken at a given state.
- calculates a penalization term dependent on k is added to the reward. This term is higher for actions that have been taken less frequently

Main features :

- simple to implement and deterministic
- might not be applicable for tasks with large state spaces as the chance of arriving in the same state again is low.

Intrinsic curiosity module
When immediate rewards are sparse it is better to use "curiosity" as an intrinsic signal controlling exploration [5]. Instead of raw data in concerns prediction $feature\ vector\ \phi(s_t)$
The method consists of two steps.

- Encode states s_t and s_{t+1} into features $\phi(s_t)$ and $\phi(s_{t+1})$ and train inverse model to predict action $\widehat{a}_t = g(s_t, s_{t+1}, \theta)$
- Using $\phi(s_t)$ and a_t to predict next $\widehat{\phi}(s_{t+1})$ with the intrinsic reward given by "curiosity"

$$r_t^i = \frac{\mu}{2}||\widehat{\phi(s_{t+1})} - \phi(s_{t+1})), \quad \text{where } \mu > 0 \text{ is the scaling factor.} \tag{2}$$

Main features:

- able to focus on the phenomena that can be either controlled or affected by the agent.
- favors transitions that have high prediction error (i.e. higher curiosity), which stimulates exploration.
- high computational complexity
- balance of intrinsic reward (curiosity) to real reward is difficult to find.
- design of intrinsic reward needs attention

References

[1] Castrignanò A, et al A methodology combining reinforcement learning and simulation to optimize the in silico culture of epithelial sheets, Journal of Computational Science, 76, 102226, 2024.

[2] Mohsen Karami et al. Revolutionizing Genomics with Reinforcement Learning Technique. In: 2023, DOI: arXiv:2302.13268

[3] Jumper, J., Evans, R., Pritzel, A. et al. Highly accurate protein structure prediction with AlphaFold. Nature 596, 583–589, 2021.

[4] Jedlicka A., Guy T.V. Exploration in Knowledge Transfer Applied to Reinforcement Learning. 2024 Submitted.

[5] Pathak, D., et al Curiosity-driven Exploration by Self-supervised Prediction. In: Proceedings of Machine Learning Research, 70, 2778-2787, 2017.