# Reinforcement Learning

## 4. Temporal difference mechanisms

Olivier Sigaud

Sorbonne Université
http://people.isir.upmc.fr/sigaud

# Temporal difference mechanisms

## Reinforcement learning

- In Dynamic Programming (planning), $T$ and $r$ are given
- Reinforcement learning goal: build $\pi^*$ without knowing $T$ and $r$
- Model-free approach: build $\pi^*$ without estimating $T$ nor $r$
- Actor-critic approach: special case of model-free
- Model-based approach: build a model of $T$ and $r$ and use it to improve the policy

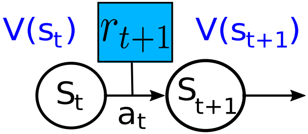
## Incremental estimation

- Estimating the average immediate (stochastic) reward in a state $s$
- $E_k(s) = (r_1 + r_2 + ... + r_k)/k$
- $E_{k+1}(s) = (r_1 + r_2 + ... + r_k + r_{k+1})/(k+1)$
- Thus $E_{k+1}(s) = k/(k+1)E_k(s) + r_{k+1}/(k+1)$
- Or $E_{k+1}(s) = (k+1)/(k+1)E_k(s) - E_k(s)/(k+1) + r_{k+1}/(k+1)$
- Or $E_{k+1}(s) = E_k(s) + 1/(k+1)[r_{k+1} - E_k(s)]$
- Still needs to store $k$
- Can be approximated as

$$E_{k+1}(s) = E_k(s) + \alpha[r_{k+1} - E_k(s)] \tag{1}$$

- Converges to the true average (slower or faster depending on $\alpha$) without storing anything
- Equation (1) is everywhere in reinforcement learning

## Temporal Difference error




- The goal of TD methods is to estimate the value function $V(s)$
- If estimations $V(s_t)$ and $V(s_{t+1})$ were exact, we would get $V(s_t) = r_{t+1} + \gamma V(s_{t+1})$
- The approximation error is

$$\delta_t = r_{t+1} + \gamma V(s_{t+1}) - V(s_t) \tag{2}$$

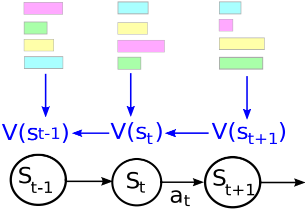
- $\delta_t$ measures the error between $V(s_t)$ and the value it should have given $r_{t+1} + \gamma V(s_{t+1})$
- If $\delta_t > 0$, $V(s_t)$ is under-evaluated, otherwise it is over-evaluated
- $V(s_t) \leftarrow V(s_t) + \alpha\delta_t$ should decrease the error (value propagation)

## Temporal Difference update rule

$$E_{k+1}(s) = E_k(s) + \alpha[r_{k+1} - E_k(s)] \quad (1)$$

$$\delta_t = r_{t+1} + \gamma V(s_{t+1}) - V(s_t) \quad (2)$$

$$V(s_t) \leftarrow V(s_t) + \alpha[r_{t+1} + \gamma V(s_{t+1}) - V(s_t)] \quad (3)$$




$$V(s_t) \leftarrow V(s_t) + \alpha[r_{t+1} + \gamma V(s_{t+1}) - V(s_t)] \quad (3)$$

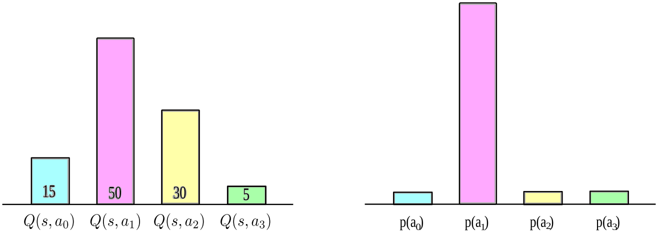
- Combines two estimation processes:
  - incremental estimation (1)
  - value propagation from $V(s_{t+1})$ to $V(s_t)$ (2)

## The Policy evaluation algorithm: TD(0)

- An agent performs a sequence $s_0, a_0, r_1, \cdots, s_t, a_t, r_{t+1}, s_{t+1}, a_{t+1}, r_{t+2}, \cdots$
- Performs local Temporal Difference updates from $s_t$, $s_{t+1}$ and $r_{t+1}$
- Proved in 1994 provided $\epsilon$-greedy exploration
- Note: updates can be performed in any order

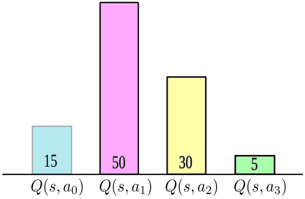
Dayan, P. & Sejnowski, T. (1994). TD(lambda) converges with probability 1. *Machine Learning*, 14(3):295–301.

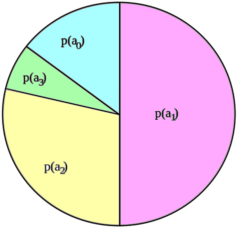
## $\epsilon$-greedy exploration




- Choose the best action with a high probability, other actions at random with low probability
- Same properties as random search
- Every state-action pair will be enough visited under an infinite horizon
- Useful for convergence proofs

## Roulette wheel

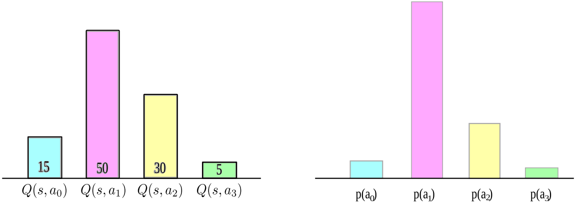



$$p(a_i) = \frac{Q(s, a_i)}{\sum_j Q(s, a_j)}$$

- The probability of choosing each action is proportional to its value

## Softmax exploration




$$p(a_i) = \frac{e^{\frac{Q(s,a_i)}{\beta}}}{\sum_j e^{\frac{Q(s,a_j)}{\beta}}}$$

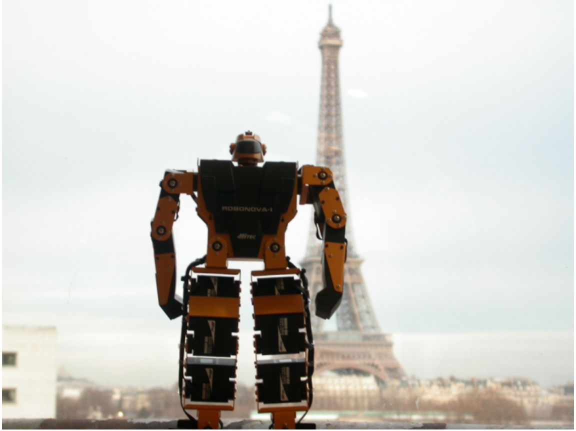
- The parameter $\beta$ is called the temperature
- If $\beta \rightarrow \infty$, all actions have the same probability $\rightarrow$ random choice
- If $\beta \rightarrow 0$, increase contrast between values
- More used in computational neurosciences
- In machine learning RL, one often uses $\tau = \frac{1}{\beta}$, i.e. $p(a_i) = \frac{e^{\tau Q(s,a_i)}}{\sum_j e^{\tau Q(s,a_j)}}$
- Meta-learning: tune $\beta$ dynamically (exploration/exploitation)

George Velentzas, Costas Tzafestas, and Mehdi Khamassi. (2017) Bio-inspired meta-learning for active exploration during

## TD(0): limitation

- TD(0) evaluates $V(s)$
- One cannot infer $\pi(s)$ from $V(s)$ without knowing $T$: one must know which $a$ leads to the best $V(s')$
- Three solutions:
  - Q-LEARNING, SARSA: Work with $Q(s,a)$ rather than $V(s)$.
  - ACTOR-CRITIC methods: Simultaneously learn $V$ and update $\pi$
  - DYNA: Learn a model of $T$: model-based (or indirect) reinforcement learning

## Any question?



Send mail to: `Olivier.Sigaud@isir.upmc.fr`

ISIR
INSTITUT DES SYSTÈMES INTELLIGENTS ET DE ROBOTIQUE

Dayan, P. and Sejnowski, T. (1994).
TD(lambda) converges with probability 1.
*Machine Learning*, 14(3):295–301.

Velentzas, G., Tzafestas, C., and Khamassi, M. (2017).
Bio-inspired meta-learning for active exploration during non-stationary multi-armed bandit tasks.
In *2017 Intelligent Systems Conference (IntelliSys)*, pages 661–669. IEEE.

INSTITUT DES SYSTÈMES INTELLIGENTS ET DE ROBOTIQUE