Reinforcement Learning (in robotics)

Petr Váňa

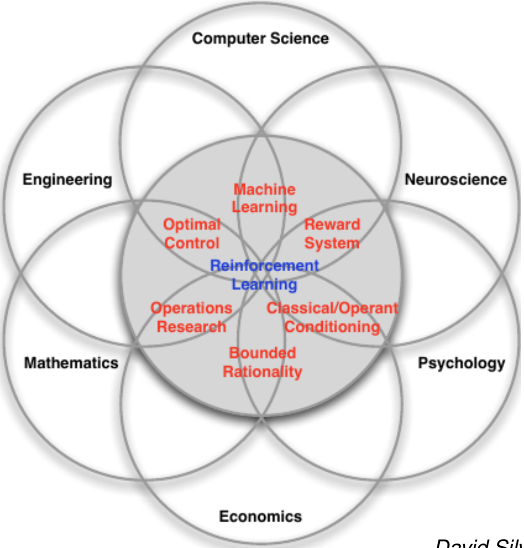
Artificial Intelligence Center
Center for Robotics and Autonomous Systems
Czech Technical University in Prague

October 29, 2016

What is Reinforcement Learning?

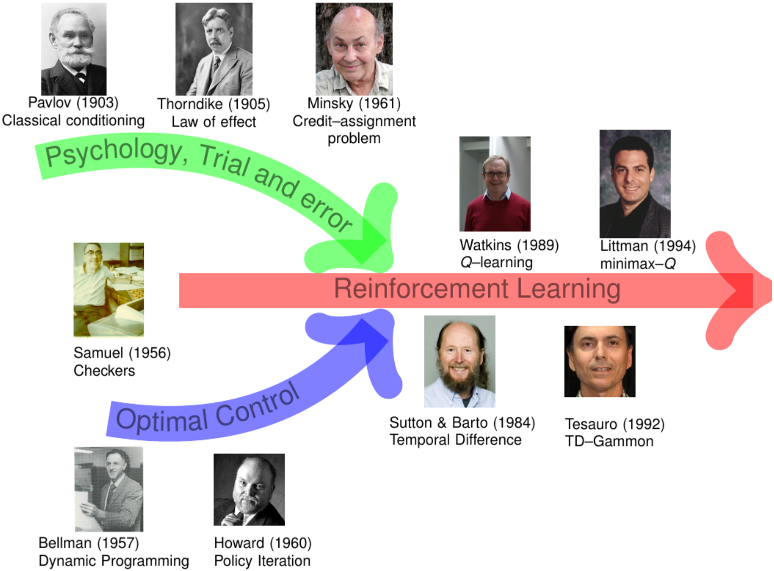
- There is no supervisor, only a **reward** signal
- Agent-oriented learning – learning by interacting with and environment to achieve a goal
  - more **realistic and ambitious** than oter kind of machine learning
- Learning by **trial and error** with delayed reward
  - most like natural learning
  - learning that can tell for itself when it is right or wrong

Reinforcement Learning (in robotics)

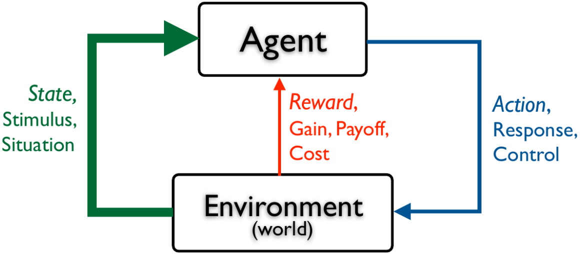


David Silver 2015

Reinforcement Learning (in robotics)



Reinforcement learning interface



- At each step t the agent:
  - Executes action a_t
  - Receives observation o_t
  - Receives scalar reward r_t
- The environment:
  - Receives action a_t
  - Emits observation o_t
  - Emits scalar reward r_t

Examples

- Defeat the world champion at Backgammon
  (Tesauro 1995)
- Learned acrobatic helicopter autopilots
  (Ng, Abbeel, Coates et al 2006+)
- Used to make strategic decisions in Jeopardy
  (IBM's Watson 2011)
- Play many different Atari games better than humans
  (Google Deepmind 2015)

Videos

The k-armed Bandit problem

- On each of an infinite sequence of time steps t you choose and action A_t from k possibilities, and receiver a real-valued reward R_t
- The reward depend only on the action taken:
  - it is indentically, independently distributed (i.i.d.)

$$q_*(a) \doteq \mathbb{E}[R_t | A_t = a], \quad \forall a \in \{1, \dots, k\}$$

- These true values are unknown. The distribution in unknown
- Nevertheless, you must maximize your total reward
- You must both try to learn their values (explore), and prefer those that apper best (exploit)

Exploration vs Exploitation Dilemma

- Online decision making involves a fundamental choice:
  - **Exploitation** – make the best decision given current information
  - **Exploration** – gather more information, try unknown

ε-greedy Action Selection

- In ε-greedy, you ar usually gready, but with probalitity ε you instad pick a random action

A simple bandit algorithm

Initialize, for $a = 1$ to k:

$Q(a) \leftarrow 0$

$N(a) \leftarrow 0$

Repeat forever:

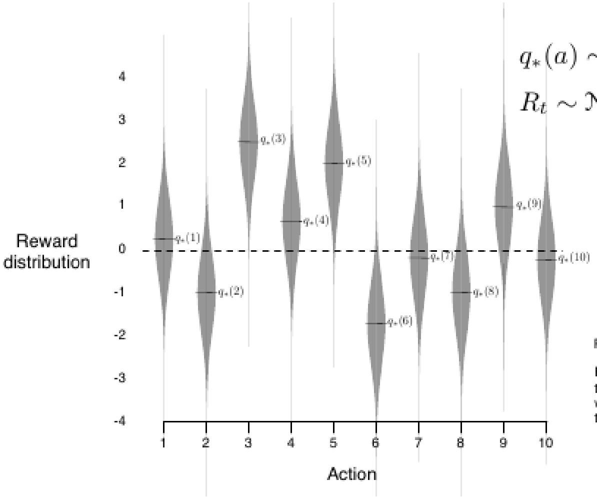
$$A \leftarrow \begin{cases} \arg\max_a Q(a) & \text{with probability } 1-\varepsilon \quad \text{(breaking ties randomly)} \\ \text{a random action} & \text{with probability } \varepsilon \end{cases}$$

$R \leftarrow bandit(A)$

$N(A) \leftarrow N(A) + 1$

$Q(A) \leftarrow Q(A) + \frac{1}{N(A)}\big[R - Q(A)\big]$

The 10-armed Testbed



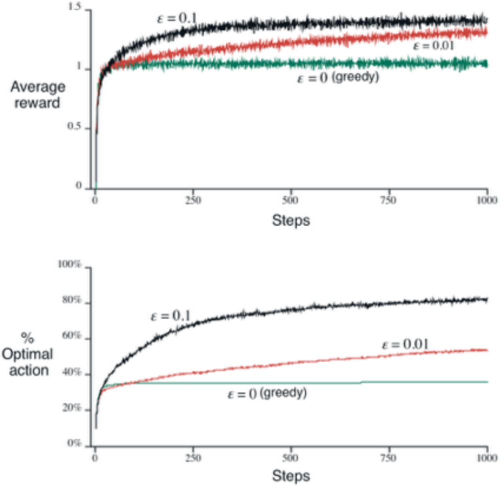
$q_*(a) \sim \mathcal{N}(0, 1)$

$R_t \sim \mathcal{N}(q_*(a), 1)$

Run for 1000 steps

Repeat the whole thing 2000 times with different bandit tasks

The 10-armed Testbed



Tracking a Non-stationary Problem

- Suppose the true action values change slowly over time
- In this case, samples averages are not a good idea
- Better is an "exponential , recency-weighted average"

$$\mathcal{Q}_{n+1} = \mathcal{Q}_n + \alpha \left[\mathcal{R}_n - \mathcal{Q}_n\right]$$

where α is a constant, step-size parameter, $0 < \alpha \leq 1$

Markov Decision Processes (MDPs)

- Markov Decision Processed formally describe an environment for reinforcement learning
- The environment is fully observable
  - the current state completely characterizes the process
- Almost all RL problems can be formalized as MDPs
  - Optimal control primarily deals with continuous MDPs
  - Partially observable problems can be converted into MDPs

Markov Assumption

Definition

A stochastic process X_t is said to be Markovian if and only if

$$\mathbb{P}(X_{t+1} = j | X_t = i, X_{t-1} = k_{t-1}, \ldots, S_1 = k_1, X_0 = k_0) = \mathbb{P}(X_{t+1} = j | X_t = i)$$

- The state captures all the information from history
- Once the state is known, the history may be thrown away
- The state is sufficient statistic for the future
- The conditional probabilities are transition probabilities

Policy Evaluation

- For a given policy π, compute teh state-value function V^{π}
- State-value function for policy π:

$$V^{\pi}(s) = \mathbb{E}\left\{\sum_{t=0}^{\infty} \gamma^{t} r_{t} | s_{0} = s\right\}$$

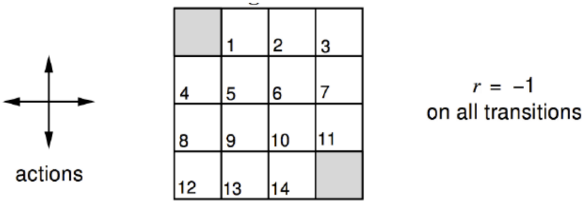
- Bellman equation for V^{π}:

$$V^{\pi}(s) = \sum_{a \in \mathcal{A}} \pi(a|s) \left[R(s,a) + \gamma \sum_{s' \in \mathcal{S}} P(s'|s,a) V^{\pi}(s')\right]$$

- Solution in matrix notation:

$$V^{\pi} = (I - \gamma P^{\pi})^{-1} R^{\pi}$$

Gridworld Example



Gridworld - Iterative evaluation

V_k for the Random Policy

$k = 0$

0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0

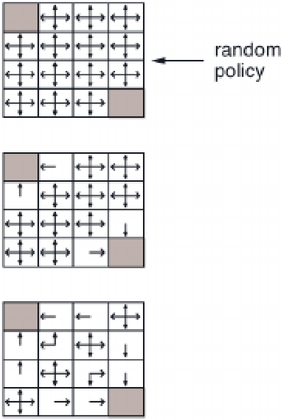
$k = 1$

0.0	-1.0	-1.0	-1.0
-1.0	-1.0	-1.0	-1.0
-1.0	-1.0	-1.0	-1.0
-1.0	-1.0	-1.0	0.0

$k = 2$

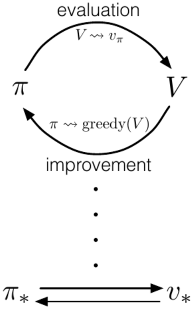
0.0	-1.7	-2.0	-2.0
-1.7	-2.0	-2.0	-2.0
-2.0	-2.0	-2.0	-1.7
-2.0	-2.0	-1.7	0.0

Greedy Policy w.r.t. V_k

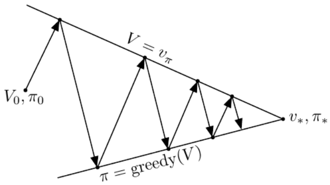


Dynamic programming-based methods

- Policy iteration
  - policy evaluation
  - policy improvement
- **Bootstrapping:** updating estimates based on other estimates
- Generaliyed Policy Iteration (GPI):

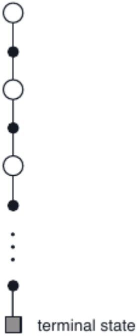


A geometric metaphor for convergence of GPI:



Monte Carlo methods

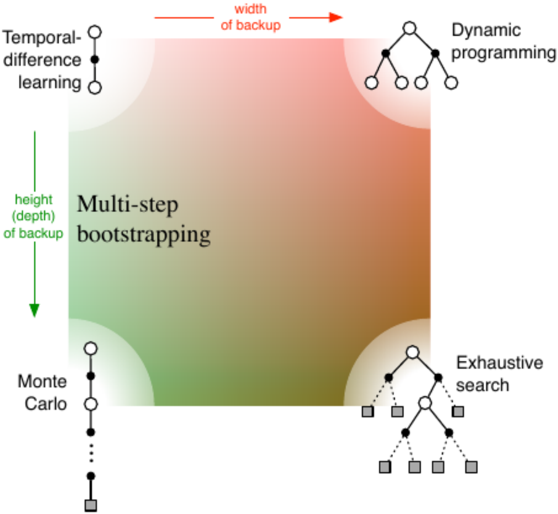
- Monte Carlo methods are learning methods
  Experience » values, policy
- MC methods can be used in two ways:
  - **Model-free**: No model necessary and still attains optimality
  - **Simulated**: Needs only a simulation, not a full model
- MC does not bootstrap from successor states's values (unlike DP)



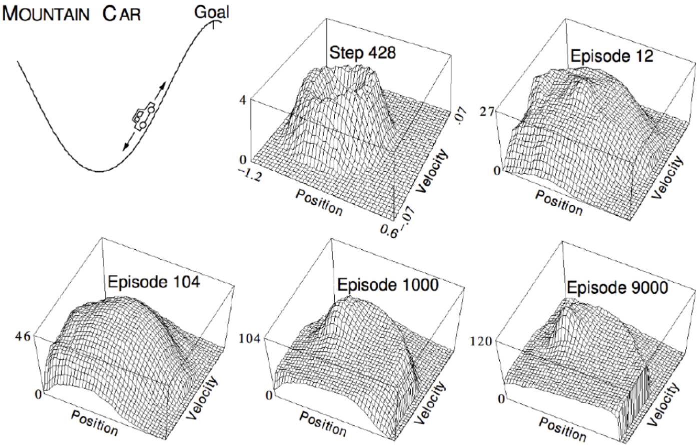
Temporal difference methods

- These methods bootstrap and sample, combining aspects of DP and MC methods
- If the world is truly Markov, then TD methods will learn faster than MC methods
- Extend prediction to control byt emlploying some form of GPI
  - On-policy control: Sarsa, Expected Sarsa
  - Off-policy control: Q-learning, Expected Sarsa

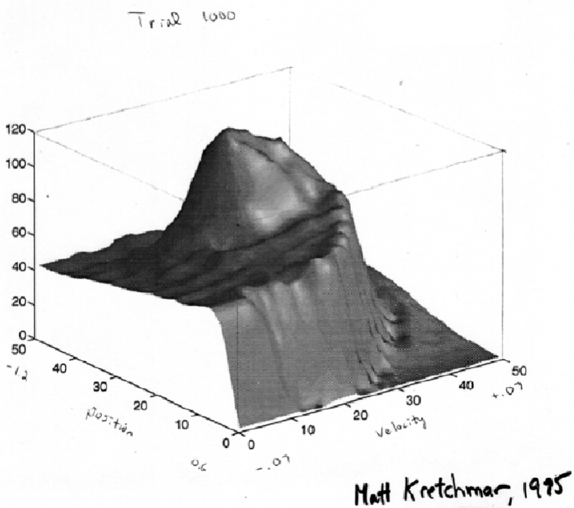
Unified View



Linear SARSA for Mountain Car



Linear SARSA for Mountain Car



Challenges in robot reinforcement learning

- Curse of dimensionality
- Curse of real-world samples
- Curse of under-modelling and model uncertainty
- Curse of goal specification

Thank You!

vanapet1@fel.cvut.cz