Reinforcement Learning Specialization - Lecture Note

Course 1 - Fundamentals of Reinforcement Learning

Module 2

New terms:

short/long-term reward

policies

planning methods

dynamic programming

reward

time steps

Video: Sequential Decision Making with Evaluative Feedback

Action-Value function

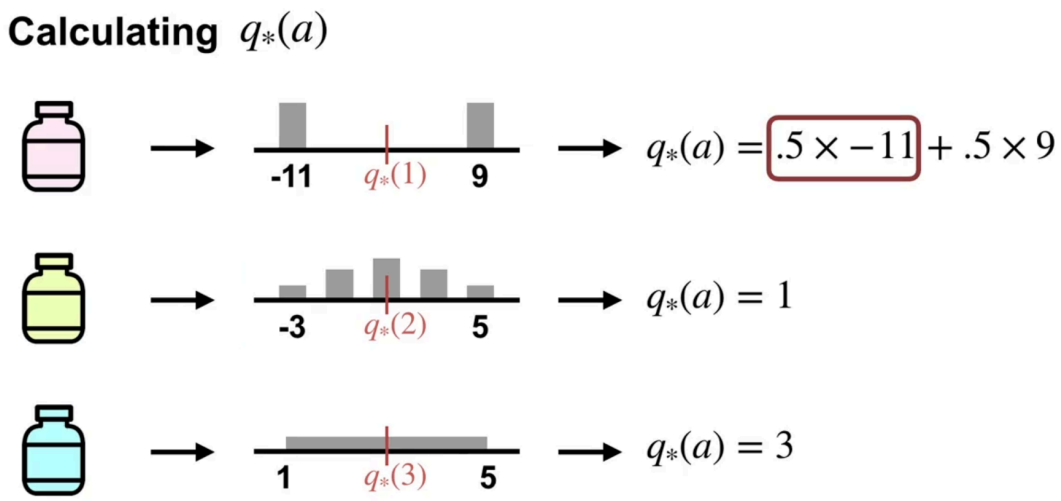
- Giá trị của hành động (q_*) là giá trị kỳ vọng của tất cả các giá trị khả thi khi thực hiện hành động *a*

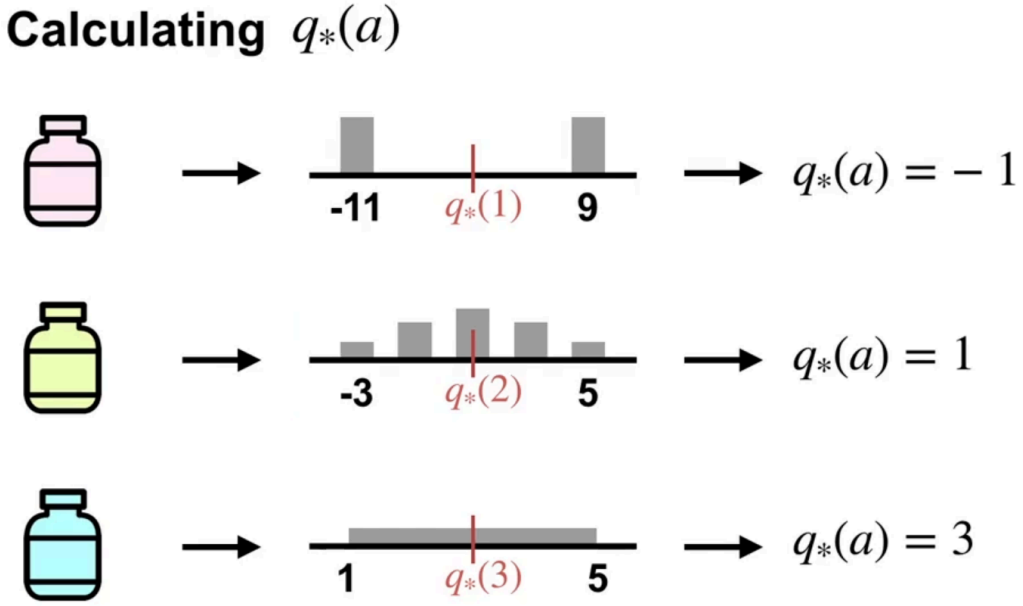
$$q_*(a) \doteq \mathrm{E}[R_t \mid A_t = a] \quad \forall a \in \{1, \dots, k\}$$
$$= \sum_r p(r \mid a)\, r$$

  Giá trị của hành động q là số chưa biết -> cần được ước tính! q_*: giá trị kỳ vọng thực sự q: giá trị kỳ vọng ước tính

- Mục tiêu là chọn hành động *a* để tối đa hóa phần thưởng/giá trị kỳ vọng của hành động

$$\arg\max_a q_*(a)$$





How is the bandit problem similar or different to the supervised learning problem?

Vietnamese

Giống: cả 2 đều có mục tiêu đạt được kết quả tối ưu được đo lường bởi 1 hàm số (supervised: loss function, bandit problem: q*/reward function), supervised được train trên 1 tập data hữu hạn và cố định, bandit problem thì có số lượng action là 1 tập hữu hạn các action (K).

Khác: supervised learning tối đa hóa hàm mất mát trên 1 tập data cố định, ko thay đổi, label là cố định; bandit problem thì có label là giá trị kỳ vọng

trên 1 phân phối xác suất.

English

Similarities

1. **Optimization Objective**
    - Both aim to optimize a measurable function:
        - *Supervised Learning*: Minimizes a **loss function** (e.g., cross-entropy, MSE).
        - *Bandit Problems*: Maximizes a **reward function** (e.g., Q*-value, expected reward).
2. **Finite Action Space**
    - Supervised learning uses a fixed, finite dataset.
    - Bandit problems assume a finite set of **K actions** (e.g., choosing between K ad variants).

Key Differences

Aspect	Supervised Learning	Bandit Problems
Data Dynamics	Static dataset with fixed labels	Dynamic, stochastic rewards from a distribution
Feedback Type	Full feedback (labels for all inputs)	Partial feedback (reward only for chosen action)
Exploration Strategy	No exploration needed (deterministic training)	Requires **exploration-exploitation trade-off** (e.g., ε-greedy, UCB)
Objective	Generalize to unseen data	Maximize cumulative reward over interactions

Video: Learning Action Values

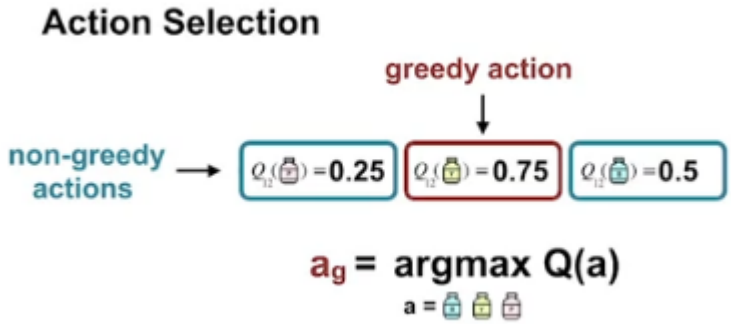
Sample-Average Method

$$Q_t(a) \doteq \frac{\text{sum of rewards when } a \text{ taken prior to } t}{\text{number of times } a \text{ taken prior to } t}$$

$$= \frac{\sum_{i=1}^{t-1} R_i}{t-1}$$

Greedy action

Method of choosing action: choosing the **greedy action** a.k.a the action currently has the largest estimated value



Video: Estimating Action Values Incrementally

Incremental update rule

Recall

$$Q_n = \frac{1}{n-1}\sum_{i=1}^{n-1} R_i$$

Incremental update rule

$$Q_{n+1} = \frac{1}{n}\sum_{i=1}^{n} R_i$$

$$= \frac{1}{n}(R_n + \sum_{i=1}^{n-1} R_i)$$

$$= \frac{1}{n}(R_n + (n-1)\frac{1}{n-1}\sum_{i=1}^{n-1} R_i)$$

$$= \frac{1}{n}(R_n + (n-1)Q_n)$$

$$= \frac{1}{n}(R_n + nQ_n - Q_n)$$

$$= Q_n + \frac{1}{n}(R_n - Q_n)$$

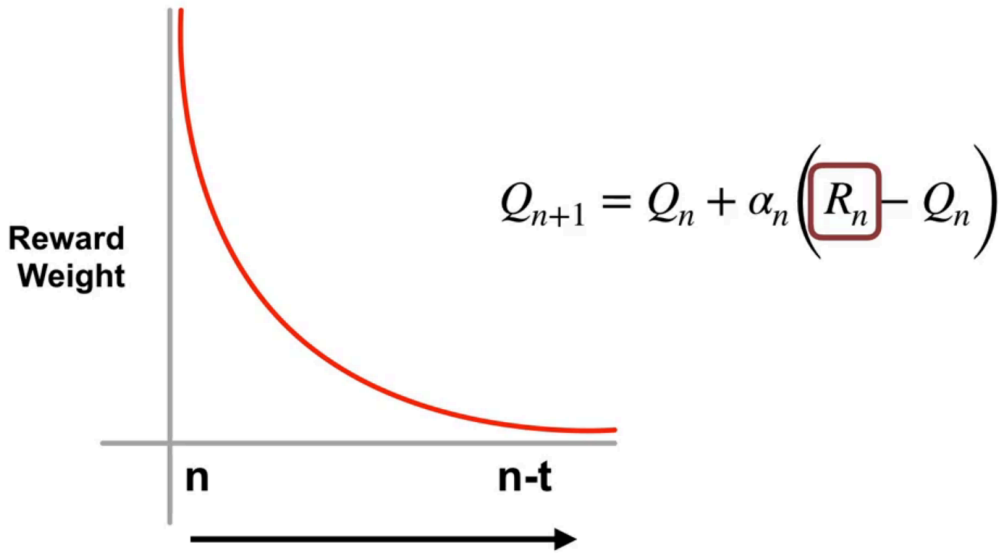
$$NewEstimate \leftarrow OldEstimate + StepSize\,[Target - OldEstimate]$$

$$Q_{n+1} = Q_n + \alpha_n (R_n - Q_n)$$
$$\alpha_n \rightarrow [0, 1]$$

Sample average method

$$\alpha_n = \frac{1}{n}$$

Non-stationary bandit problem (rewad ko cố định và có thể thay đổi theo thời gian)



the most recent rewards affect the estimate more than older rewards (các reward mới nhất ảnh hưởng đến giá trị ước lượng hơn các reward ở các bước xa hơn)

Decaying past rewards

$$
\begin{aligned}
Q_{n+1} &= Q_n + \alpha_n (R_n - Q_n) \\
&= \alpha_n R_n + Q_n - \alpha_n Q_n \\
&= \alpha_n R_n + (1 - \alpha_n) Q_n \\
&= \alpha_n R_n + (1 - \alpha_n) \left[\alpha_n R_{n-1} + (1 - \alpha_n) Q_{n-1}\right] \\
&= \alpha_n R_n + (1 - \alpha_n) \alpha_n R_{n-1} + (1 - \alpha_n)^2 Q_{n-1} \\
&= \alpha_n R_n + (1 - \alpha_n) \alpha_n R_{n-1} + (1 - \alpha_n)^2 \alpha_n R_{n-2} + \cdots \\
&\quad + (1 - \alpha_n)^{n-1} \alpha_n R_1 + (1 - \alpha_n)^n Q_1 \\
&= (1 - \alpha)^n Q_1 + \sum_{i=1}^{n} \alpha (1 - \alpha)^{n-i} R_i
\end{aligned}
$$

$$
Q_1 \rightarrow \textbf{initial action-value}
$$

Đóng góp của Q_1 với giá trị ước tính tại bước $n + 1$ giảm dần theo cấp lũy thừa theo thời gian, các giá trị reward ở các bước cũ đóng góp theo cấp lũy thừa ít hơn. Sự ảnh hưởng của giá trị khởi tạo (Q_1) tiến gần về 0 khi càng có thêm nhiều data, các giá trị mới nhất quyết định giá trị ước tính hiện tại (Q_{n+1})

Exploration vs. Exploitation Tradeoff

Video: What is the trade-off?

Exploration and Exploitation

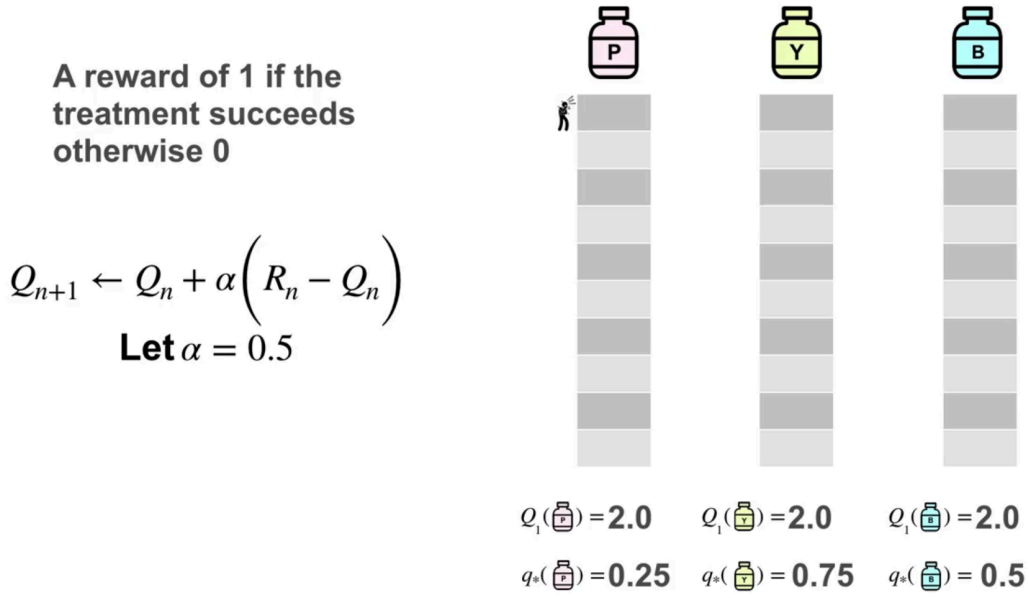
- Exploration - **improve** knowledge for **long-term** benefit
- Exploitation - **exploit** knowledge for **short-term** benefit (being greedy w.r.t estimated values, may not actually get the most reward)
- Round Robin fashion: tuần tự theo chu kỳ

Epsilon-Greedy Action Selection

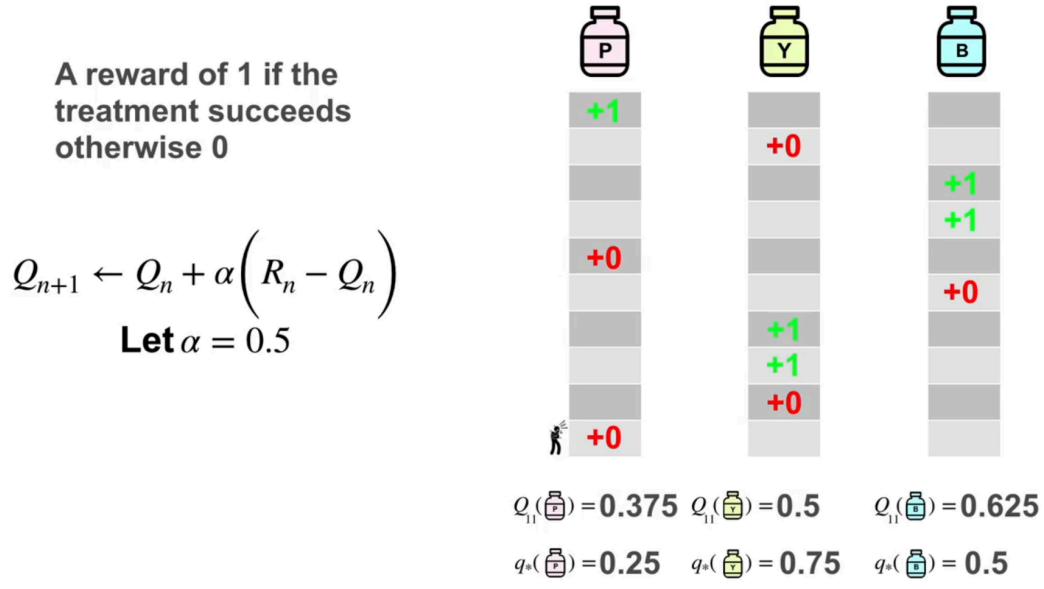
$$
A_t \leftarrow \begin{cases} \arg\max_a Q_t(a) & \text{with probability } 1 - \epsilon \\ a \sim \text{Uniform}(\{a_1, \ldots, a_k\}) & \text{with probability } \epsilon \end{cases}
$$

Phương pháp để cân bằng giữa Exploration và Exploitation

Video: Optimistic Initial Values



Khởi tạo giá trị kỳ vọng ước lượng khởi đầu cao và thực hiện chiến thuật lựa chọn tham lam (greedy selection) giúp agent có thể explore các lựa chọn khác nhau ở các timestep đầu tiên và update dần về giá trị hành động thực tế

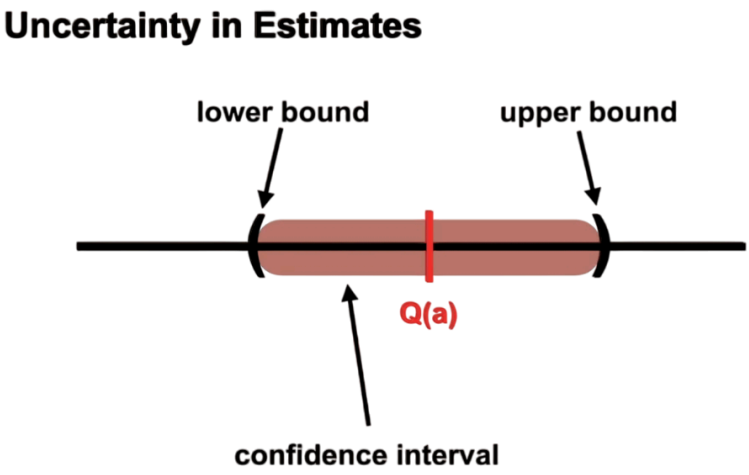


Giới hạn:

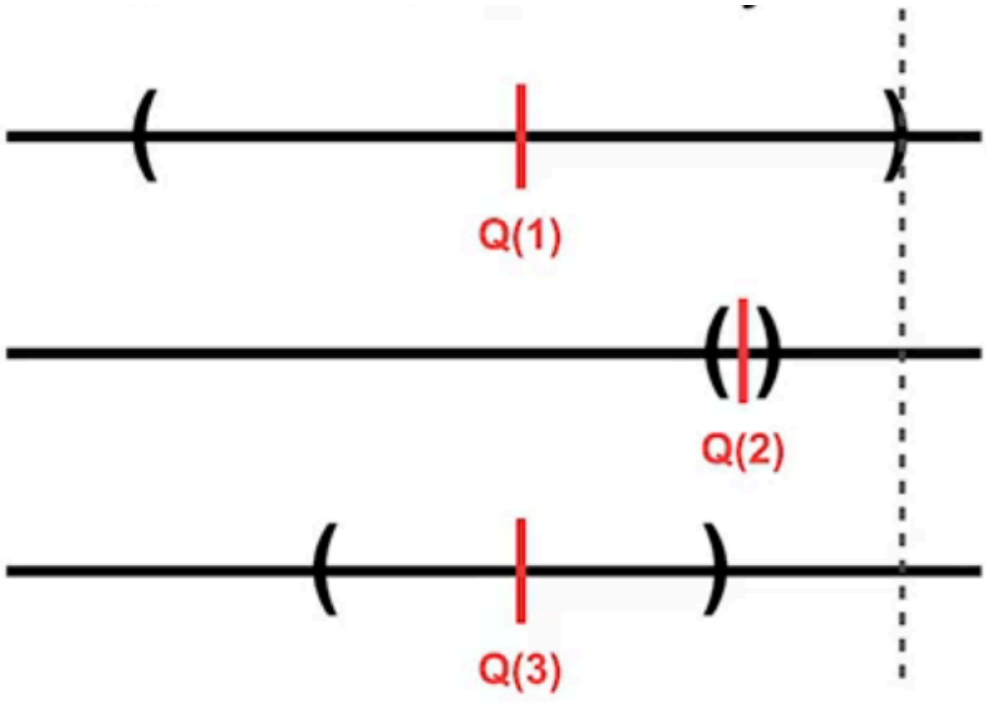
- Chỉ explore ở các bước đầu tiên, sau khi đến bước nào đó sẽ dừng explore

- Ko phù hợp với các bài toán có reward thay đổi theo thời gian (non-stationary problems)
- Không thể biết được giá trị khởi đầu lạc quan nên để là bao nhiêu (vì không biết giá trị tối đa của reward)

Video: Upper-Confidence Bound (UCB) Action Selection



Optimism in the Face of Uncertainty



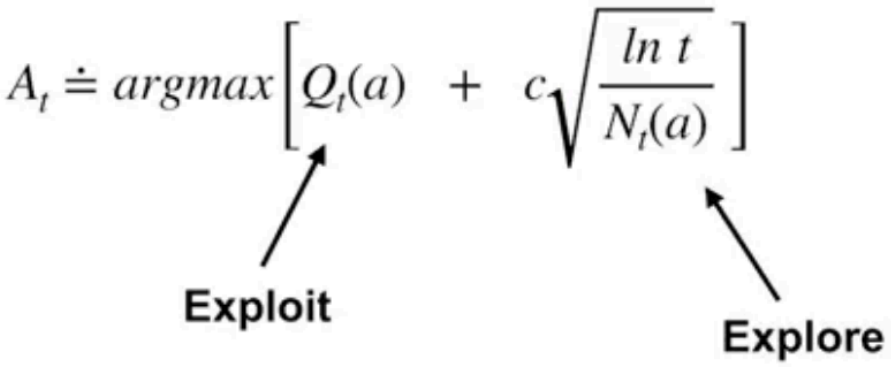
Upper-Confidence Bound (UCB) Action Selection

chọn action có cận trên của giá trị hành động là cao nhất□□□

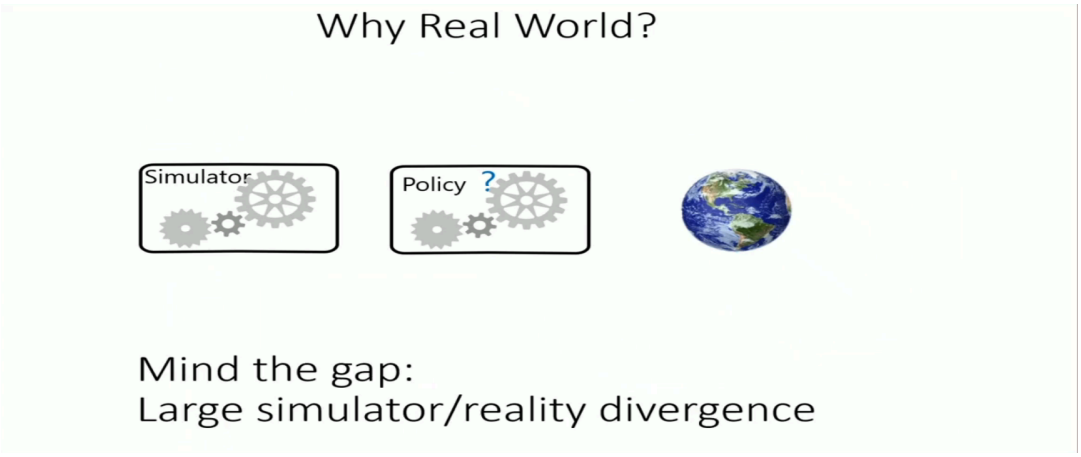
$$A_t \doteq \arg\max_a \left[Q_t(a) + c\sqrt{\frac{\ln t}{N_t(a)}} \right]$$

$c\sqrt{\frac{\ln t}{N_t(a)}}$ is upper-confidence bound (UCB) exploration term

c is user-specified parameter that controls the amount of exploration

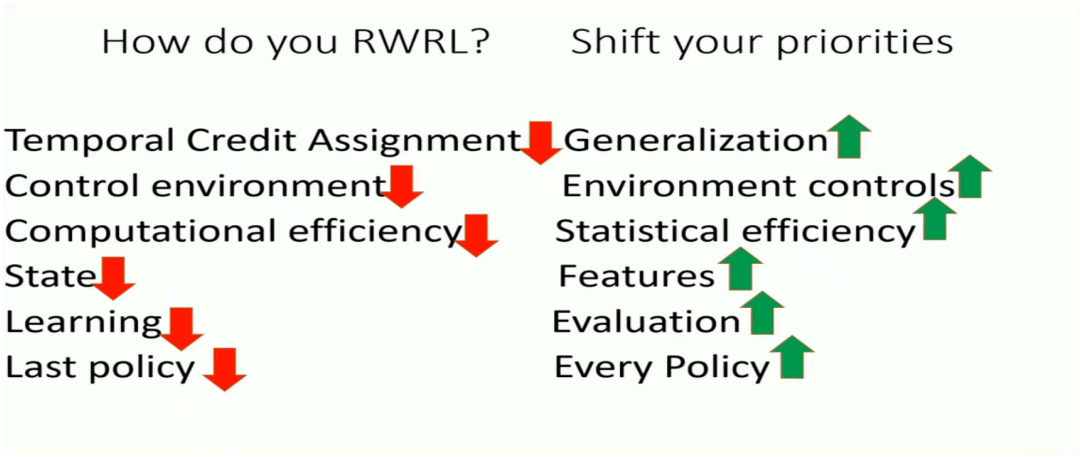


Video: Jonathan Langford: Contextual Bandits for Real World Reinforcement Learning



There's is a gap between the simulator and the reality.

Real World Reinforcement Learning


How do you RWRL?
Shift your priorities
Temporal Credit Assignment
Control environment
Computational efficiency
State
Learning
Last policy
Generalization
Environment controls
Statistical efficiency
Features
Evaluation
Every Policy