Exploration Strategies: Motivation & Multi-Armed Bandits

Christopher Mutschler

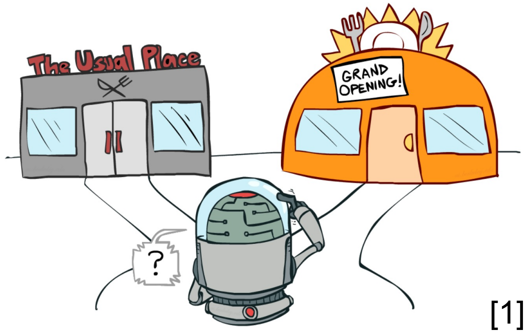
Agenda

- Motivation, Problem Definition & Multi-Armed Bandits
- Classic Exploration Strategies
  - Epsilon Greedy
  - (Bayesian) Upper Confidence Bounds
  - Thomson Sampling
- Exploration in Deep RL:
  - Count-based Exploration: Density Models, Hashing
  - Prediction-based Exploration:
    - Forward Dynamics
    - Random Networks
    - Physical Properties
  - Memory-based Exploration:
    - Episodic Memory
    - Direct Exploration
- Summary and Outlook

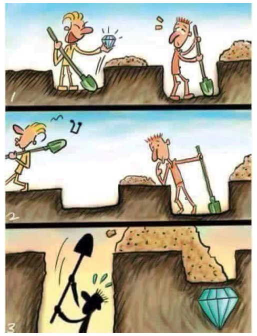
Agenda

- **Motivation, Problem Definition & Multi-Armed Bandits**
- Classic Exploration Strategies
  - Epsilon Greedy
  - (Bayesian) Upper Confidence Bounds
  - Thomson Sampling
- Exploration in Deep RL:
  - Count-based Exploration: Density Models, Hashing
  - Prediction-based Exploration:
    - Forward Dynamics
    - Random Networks
    - Physical Properties
  - Memory-based Exploration:
    - Episodic Memory
    - Direct Exploration
- Summary and Outlook

Problem Motivation: Exploration in Life



[1]



[2]

[3]

Restaurant Selection | Oil Drilling | Online Ad Placement

exploit:

go to your favorite restaurant | drill at the best-known location | show most successful ads

vs.

explore:

try something new | drill at a new location | show a different random ad

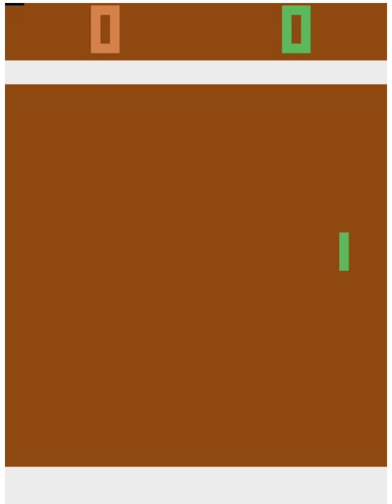
[1] Berkeley AI course
[2] https://medium.com/deep-math-machine-learning-ai/ch-12-1-model-free-reinforcement-learning-algorithms-monte-carlo-sarsa-q-learning-65267cb8d1b4
[3] https://designrshub.com/2012/05/3-smart-advertising-tips-for-an-effective-ad-placement.html

Taken from David Silver's Lecture on XX.

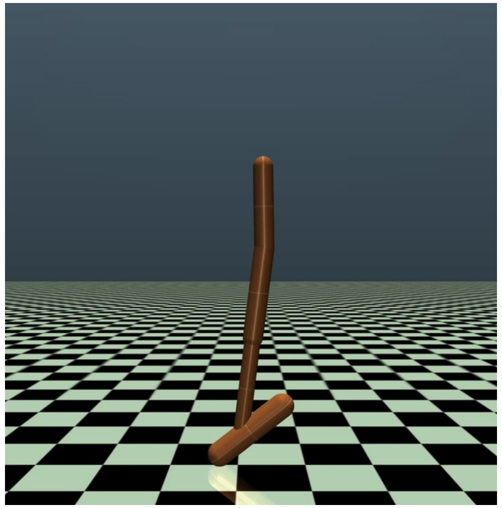
Problem Motivation: RL so far

- Improving the policy with $\pi'(s) = \underset{a \in \mathcal{A}}{\operatorname{argmax}} Q^{\pi}(s, a)$ poses problems for bootstrapping the Q-function
- We used ε-greedy policy improvement
  → occasionally try something "suboptimal" (at least we think it is)

[1]



[2]

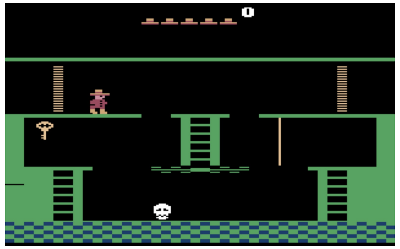


[3]

[1] https://www.youtube.com/watch?v=V1eYniJ0Rnk
[2] https://towardsdatascience.com/tutorial-double-deep-q-learning-with-dueling-network-architectures-4c1b3fb7f756
[3] https://ljvmiranda921.github.io/projects/2018/09/14/pfn-internship/

Problem Motivation: RL so far

- Oops, I forgot to tell you:
  - ε-greedy exploration does not work well on many tasks and even fails for some of them!
- Some of the Atari 2600 series games known for their hard exploration:



Montezuma's Revenge

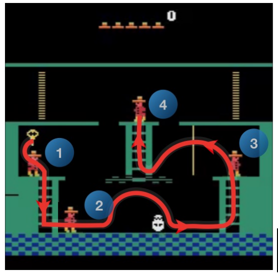
Private Eye

Seaquest

Pitfall!

Why?

- Getting key = opening door → reward
- Getting killed by skull → nothing
- ➢ *Finishing the game only weakly correlates with reward structure of the game!*

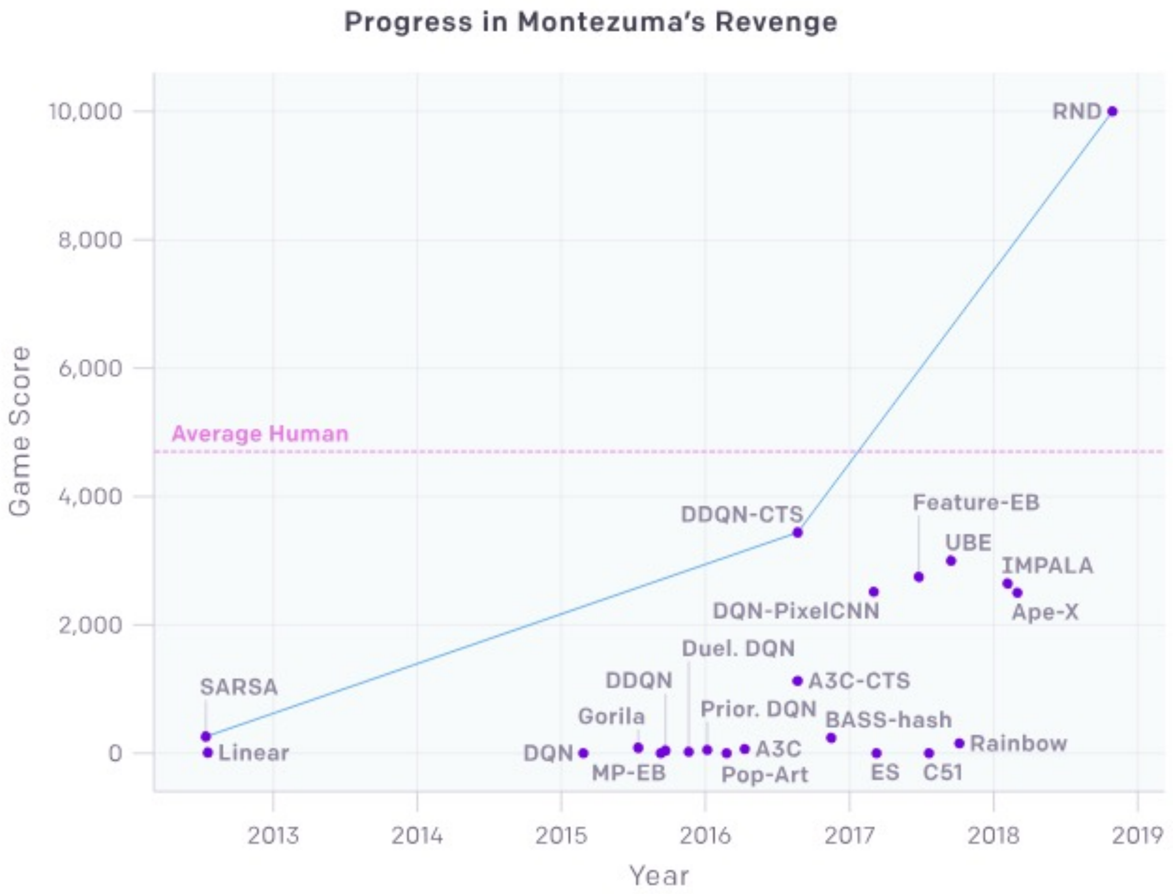


→ [1]

[1] Aytar et al.: Playing Hard Exploration Games by Watching Youtube. NeurIPS 2018.

Problem Motivation

- But: there is a solution to this – spoiler!

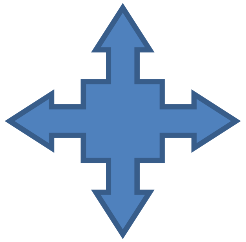


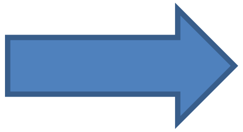
OpenAI Blog. Reinforcement Learning with Prediction-Based Rewards. October 31, 2018.

Problem Definition

- There are two potential definitions of the exploration problem:
  1. How can an agent **discover** high-reward strategies that require a temporally extended sequence of complex behaviors that, individually, are not rewarding?
  2. How can an agent **decide** whether to attempt new behaviors (to discover ones with higher reward) or continue to do the best thing it knows so far?

- Both definitions stem from the same problem:
  - **Exploration**: do things you haven't done before (in the hopes of getting even higher reward)
    → increase knowledge
  - **Exploitation**: do what you know to yield highest reward
    → maximize performance based on knowledge





See also Sergey Levine's Lecture CS285: Exploration.

Problem Definition

- The dilemma comes from *incomplete* information:
  - we need to gather enough information to make best overall decisions,
  - … while keeping the risk under control!
- With exploitation we take advantage of the best option we know
- With exploration we take risks to learn about unknown options.
- The best long-term strategy may involve short-term sacrifices

- Ok, we got it. Exploration can be very hard…
- But: how can we derive an **optimal** exploration strategy?
  - Mathematically: what does *optimal* even mean?
  - In online learning we use the term “regret” to express this (we will come to this later)

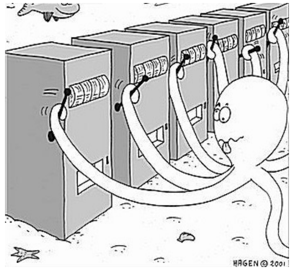
Multi-armed bandits (1-step stateless RL problems)	Contextual bandits (1-step RL problems)	Small, finite MDPs (e.g., tractable planning, model-based RL)	Large, infinite MDPs (e.g., continuous spaces)

theoretically tractable ⟷ theoretically intractable

(illustration adapted from Sergey Levine’s CS285 class from UC Berkeley)

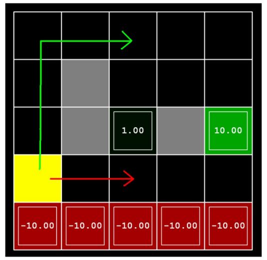
Problem Definition

- How can an exploration problem be made tractable?



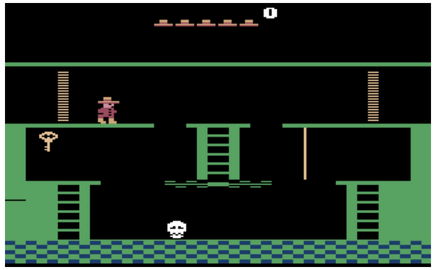
Multi-armed bandits
Contextual bandits

- Exploration problem can be formalized as POMDP identification
- Then policy learning is then easy (even with POMDP)



Small & finite MDPs

- We can frame the exploration problem as a Bayesian model identification
- Then reason about value of information



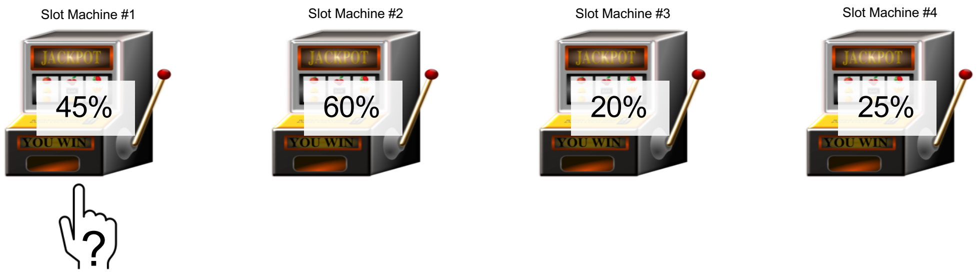
Large & infinite MDPs

- Optimal methods do not work here
- We need to take them as inspiration, or we use hacks

See also Sergey Levine's Lecture CS285: Exploration.

Multi-Armed Bandits

- The multi-armed-bandit problem is a classic problem used to study the exploration vs. exploitation dilemma
- Imagine you are in a casino with multiple slot machines, each configured with an unknown reward probability:



https://www.gameroomshow.com

- Under the assumption of an infinite number of trials:

→ ***What is the best strategy to achieve highest long-term rewards?***

Naive Solution:

1. Play each machine for many many many rounds
2. Estimate *true* reward probability of each machine (law of large numbers)
3. Act greedily with respect to the uncovered probabilities

Multi-Armed Bandits

POMDP interpretation:
this is the state, but we don't know it
- solving this yields the optimal exploration
- we could maintain a belief over the state (prob-distr. over the states → huge)

A Bernoulli multi-armed bandit can be described as a tuple of $\langle \mathcal{A}, \mathcal{R} \rangle$, where:

- We have N machines and their associated reward probabilities $\{\theta_1, \dots, \theta_n\}$
- At each time step t we take an action a_t on a single slot machine and receive a reward r_t
- $\mathcal{A}$ is a set of actions (i.e., arms): $\mathcal{A} = \{\text{pull}_1, \text{pull}_2, \dots, \text{pull}_n\}$
  - Each action refers to the interaction with one slot machine
    → the true value of the action a is the expected reward $Q(a) = \mathbb{E}[r|a] = \theta$
  - If action a_t at the time step t is on the i-th machine, then $Q(a_t) = \theta_i$ (note: value function is unknown!)
- $\mathcal{R}$ is a reward function:
  - We observe a reward r in a stochastic fashion. At the time step t, $r_t = \mathcal{R}(a_t) = p(r|a)$
    → returns reward 1 with a probability of $\theta_i = Q(a_t)$, or 0 otherwise (i.e., with probability $1 - \theta_i$).
  - The distribution $p(r|a)$ is fixed, but unknown

- Goal: maximize cumulative reward $\sum_{t=1}^{T} r_t$
- As usual, $p(a|r)$ is unknown but we still want to estimate $Q(a)$

→ This is a simplified MDP (as there are no states)

Regret

- Our goal is to maximize the cumulative reward $\sum_{t=1}^{T} r_t$
- The optimal reward probability θ^* of the optimal action a^* is

$$\theta^* = Q(a^*) = \max_{a \in \mathcal{A}} Q(a) = \max_{1 \leq i \leq K} \theta_i = \max_{a \in \mathcal{A}} \mathbb{E}[r_t | a_t = a]$$

- But how can we reason about the exploration-exploitation trade-off?
  → Regret as a *one-step opportunity loss*

- Our loss function is the total regret we might have by not select the optimal action up to the time step T:

what we did

$$\mathcal{L}_T = \mathbb{E}\left[\sum_{t=1}^{T} \left(\theta^* - Q(a_t)\right)\right] = \sum_{a \in \mathcal{A}} N_T(a)\Delta_a$$

per-action regret

what we should have been doing

action-selection counter

Regret

- If we knew the optimal action with the best reward, then:
  - Maximize cumulative rewards ≡ minimize total regret
  - The agent cannot observe or sample the real regret directly
  - But we can use it to analyze different exploration strategies!
- Note:
  - The sum for the total regret extends beyond (single step) episodes
  - The view extends over "lifetime of learning", rather than over "current episode"
  - **A good algorithm ensures small visitation counts for large action regrets** *(but action regrets are unknown…)*

- From here, we can derive 3 different **bandit strategies**:
  1. No exploration: very naïve approach and a bad one usually
  2. Exploration at random
  3. Smart exploration with preference to explore actions with high uncertainty