Exploration and exploitation

Exploration and exploitation

Ostbayerische Technische Hochschule Amberg-Weiden

Exploration vs. exploitation dilemma

Online decision-making involves a fundamental choice:

- Exploitation: Make the best decision given current information
- Exploration: Gather more information

Consequences

- The best long-term strategy may involve short-term sacrifices
- Only gather as much information as necessary to make the best overall decision

Four methods are presented

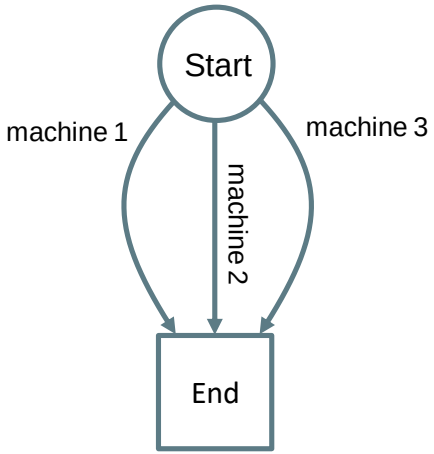
- optimistic initialization
- R-Max
- decaying ϵ-greedy
- UCB1

Exploration and exploitation

Ostbayerische Technische Hochschule Amberg-Weiden

Example: Multi-armed bandits

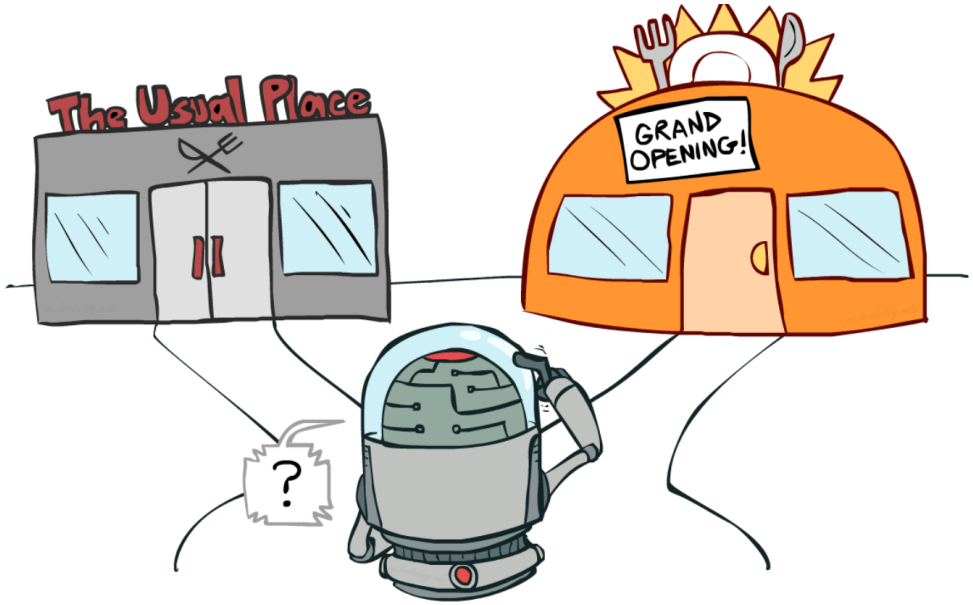
- Three actions for a given state: Pull lever of slot machine 1/2/3 and get reward
- Special case: Probabilistic (non-deterministic) rewards R
- At time t, agent selects action a_t and gets reward R_t
- Goal: Maximize (average) reward R_t



Exploration and exploitation

Related real life examples

- Should I stick to my current job or get a new one?
- How many prices should I compare before buying product X?
- Secretary problem



http://procaccia.info/courses/15381f16/slides/781_rl_rmax.pdf

Task: What other related examples come to your mind?

Exploration and exploitation

Ostbayerische Technische Hochschule Amberg-Weiden

Disclaimer: All methods described in this chapter are only thoroughly analyzed for multi-armed bandits and not for MDPs in general, although they also provide very good (but not optimal) results for MDPs with discrete actions

Difference between multi-armed bandit and general MPD

	Multi-armed bandit	MDP
episode length	1	usually > 1
reward	stochastic	usually deterministic
goal	max. (avg.) reward	max (avg.) discounted reward

Exploration and exploitation

Ostbayerische Technische Hochschule Amberg-Weiden

Regret

- Is based on the optimal V-value, defined as $V_* = \max_a Q_*(a)$
- Defines how much worse an action is compared to the optimal action at time step t

$$I_t = E[\, V_* - Q(A_t)\,]$$

Total regret

- Is the accumulated regret until timestep t

$$L_t = E\left[\sum_{\tau=1}^{t} V_* - Q(A_\tau)\right]$$

- Maximize return $\widehat{=}$ minimize total regret
- always executing the best action equals total regret of 0

Exploration and exploitation

Ostbayerische Technische Hochschule Amberg-Weiden

Counting regret

- the total regret depends on two factors
  - how bad an action is
    (**gap** $\Delta_a = V_* - Q(a)$ as difference of action a to optimal action)
  - how often the action is taken
    (count $N_t(a)$ as the number of selecting a until timestep t)
- the counting regret is a reformulation of the total regret as

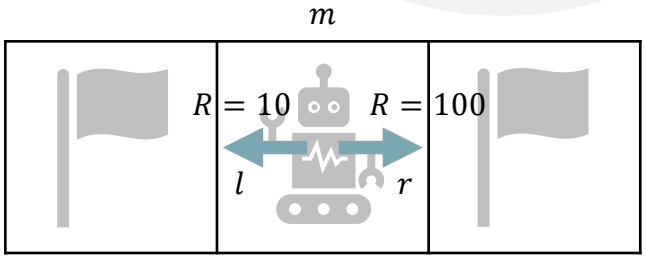
$$L_t = E\left[\sum_{\tau=1}^{t} V_* - Q(A_\tau)\right]$$

← iterate over time steps τ

$$= \sum_{a \in \mathcal{A}} E[N_t(a)] \cdot \Delta_a$$

← iterate over actions a

Exploration and exploitation



Example: Trivial robot maze

- Policy: Move right for two episodes, then move left for one episode.

$$V_*(m) = 100$$
$$Q(m, r) = 100 \quad \rightarrow \Delta_r = 0$$
$$Q(m, l) = 10 \quad \rightarrow \Delta_l = 90$$

- After n timesteps/episodes with counting regret

$$E[N_n(r)] = \frac{2}{3}n \quad \rightarrow \quad L_n = \frac{0 \cdot 2}{3}n + \frac{90 \cdot 1}{3}n = 30n$$
$$E[N_n(l)] = \frac{1}{3}n$$

- After n timesteps/episodes with total regret

$$L_n = E[\, 0 + 0 + 90 + 0 + 0 + 90 + \cdots \,] = 30n$$

Exploration and exploitation

Total regret for different policies

- Optimal policy: Always pick the best action

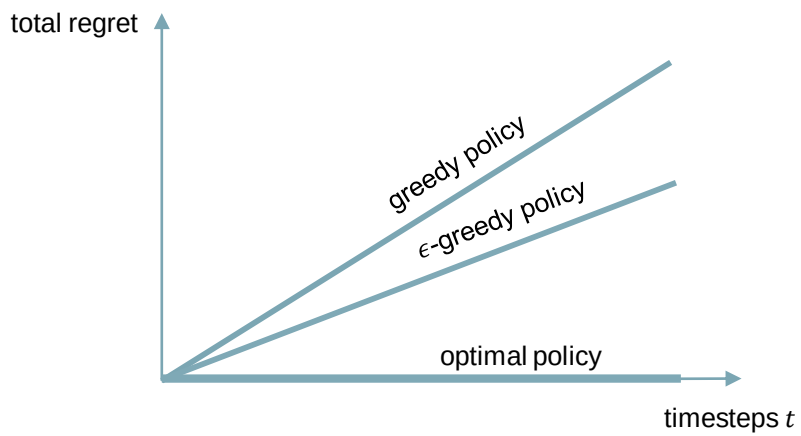
$$L_t = 0$$

- greedy policy: Sometimes (systematically) pick the wrong action

$$L_t = c \cdot t$$

- ϵ-greedy policy: With probability ϵ pick the wrong action

$$L_t = d \cdot t$$



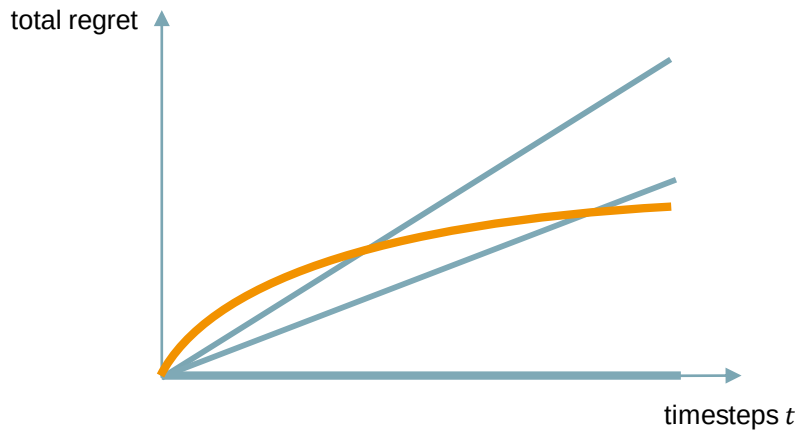
Exploration and exploitation

Other considerations

- If an algorithm explores forever, it will have a linear total regret
- If an algorithm explores never, it will have a linear total regret
- If an algorithm stops exploration too early, it will have a linear total regret

Are there exploration algorithms with sublinear total regret converging to an optimal policy?

→ yes



Exploration and exploitation

Ostbayerische Technische Hochschule Amberg-Weiden

Total regret lower bound

- defines how small the total regret can become
- depends on the reward distribution $\mathcal{R}^a$ when picking action a and the **Kullback-Leibler divergence KL** measuring the dissimilarity between two probability distributions
- Theorem (Lai and Robbins)

$$\lim_{t\to\infty} L_t \geq \log t \cdot \sum_{a|\Delta_a>0} \frac{\Delta_a}{KL(\mathcal{R}^a|\mathcal{R}^{a^*})}$$

the larger the gaps are, the worse it is on average if the wrong action is picked

most important: total regret increases logarithmically over t in the best case

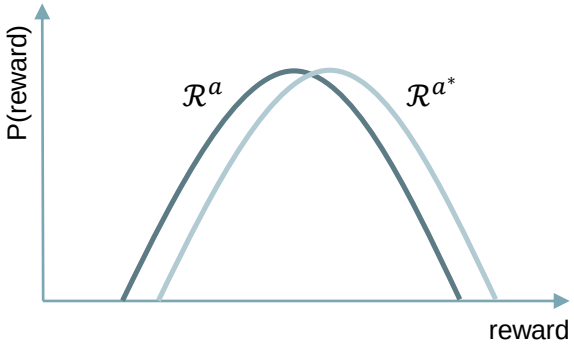
sum over all actions with a gap > 0 (that are the suboptimal actions)

the smaller the KL divergence is, the more similar the reward distributions are and the more difficult it becomes to differentiate the reward distributions

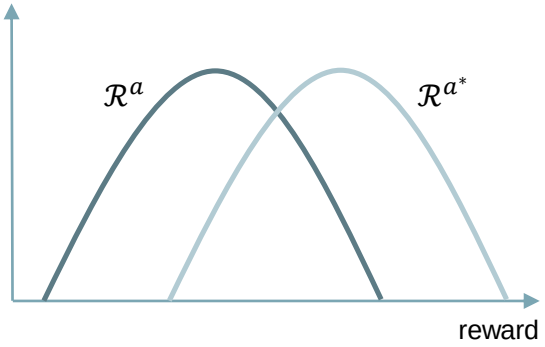
Exploration and exploitation

Ostbayerische Technische Hochschule Amberg-Weiden

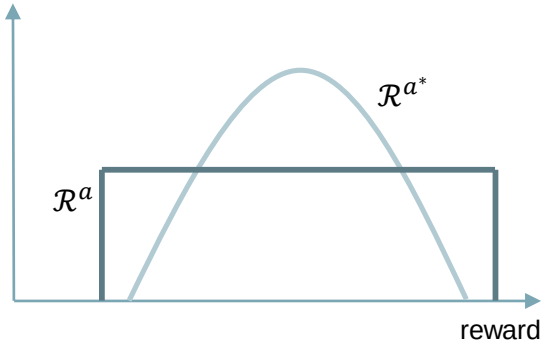
Example



small KL div



large KL div



large KL div

Task: Create different probability distributions and estimate the KL divergence

Exploration and exploitation

Optimism in the face of uncertainty

- Is a heuristic/idea how to behave if one has to decide between multiple actions with unknown reward
- Colloquially: "If you have no idea about the outcome of an action, pick it. It might be the best"

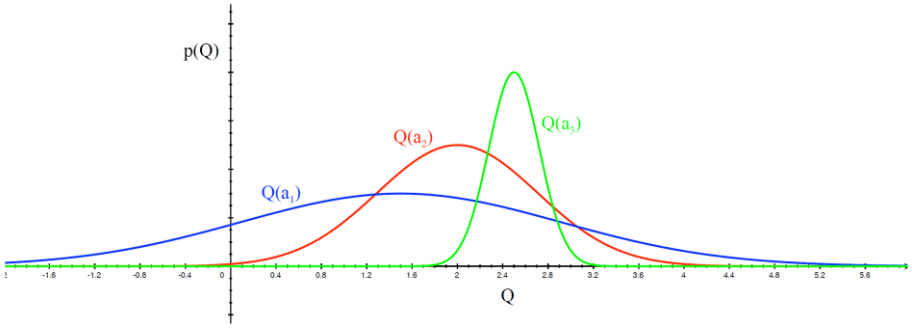
https://medium.com/@uzair.ahmed2/optimism-a-path-to-success-4c7c4cc23fd7

Exploration and exploitation

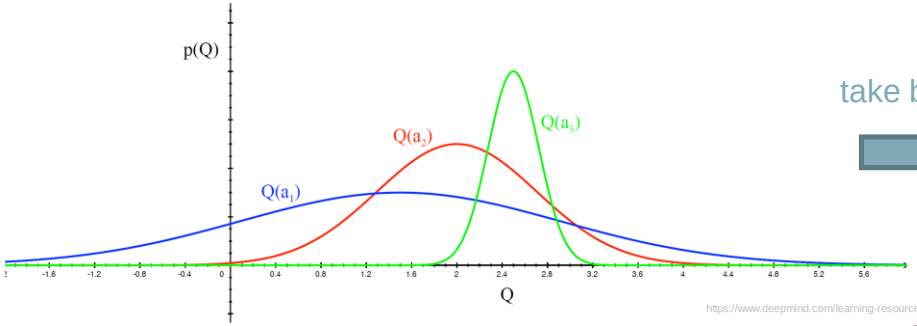
Ostbayerische Technische Hochschule Amberg-Weiden

Example: Estimated reward distributions for three actions

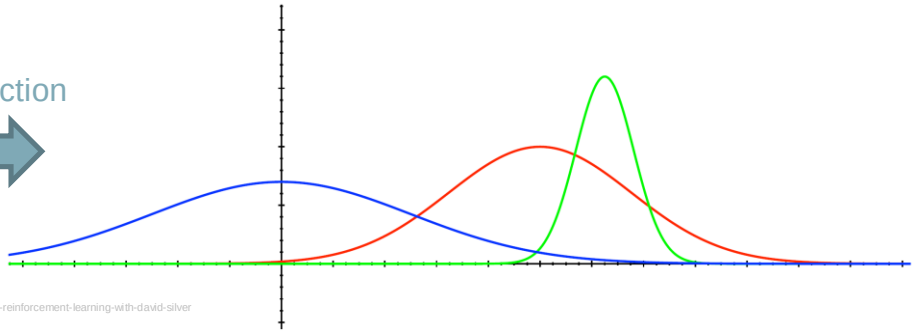
- A greedy policy would take the green action
- An optimistic policy would take the blue action



- After taking the blue action, one is more certain about the outcome of the blue action
  → might take another one next time



take blue action



Exploration and exploitation

Ostbayerische Technische Hochschule Amberg-Weiden

Optimistic initialization

- Initialize Q-function to a high value
- Then update it using MC evaluation / SARSA / Q-learning
  (SARSA / Q-learning may take long to converge if initial value is too large)
- Linear total regret when being combined with (ϵ-)greedy
  (no guarantee that optimal action will be taken after some time in every state)
- Standard for a lot of practical problems

Q-learning (off-policy TD control) for estimating $\pi \approx \pi_*$

Algorithm parameters: step size $\alpha \in (0, 1]$, small $\varepsilon > 0$
Initialize $Q(s, a)$, for all $s \in \mathcal{S}^+, a \in \mathcal{A}(s)$, arbitrarily except that $Q(terminal, \cdot) = 0$

Loop for each episode:
  Initialize S
  Loop for each step of episode:
    Choose A from S using policy derived from Q (e.g., ε-greedy)
    Take action A, observe R, S'
    $Q(S, A) \leftarrow Q(S, A) + \alpha\big[R + \gamma \max_a Q(S', a) - Q(S, A)\big]$
    $S \leftarrow S'$
  until S is terminal

Sutton, Barto: Reinforcement Learning

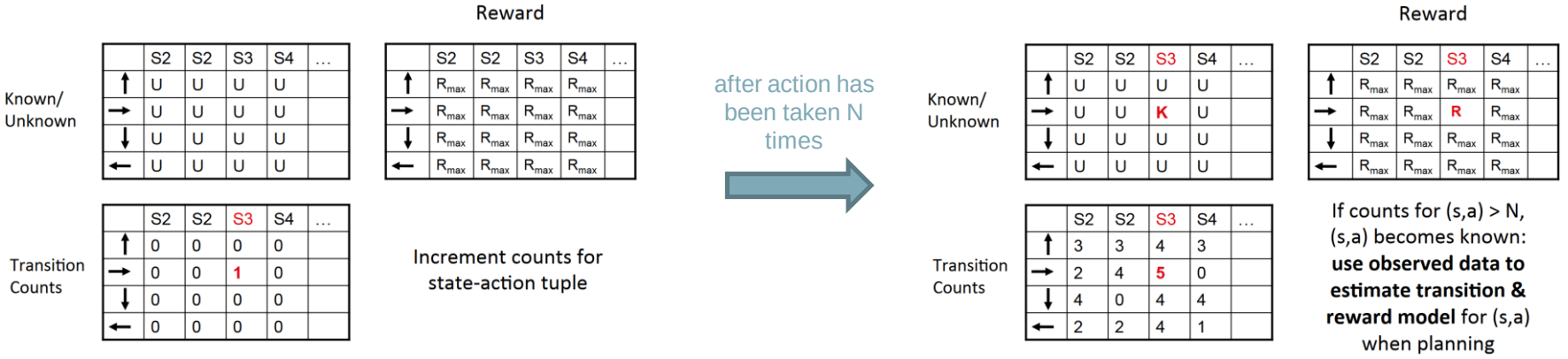
Exploration and exploitation

Ostbayerische Technische Hochschule Amberg-Weiden

R-max

- Initialize rewards R to the maximum value R_{max}
- If the corresponding action has been taken N times, replace R_{max} with observed reward
- Linear total regret when being combined with (ϵ-)greedy
  (no guarantee that optimal action will be taken after some time in every state)

Example



http://procaccia.info/courses/15381f16/slides/781_rl_rmax.pdf

Exploration and exploitation

Ostbayerische Technische Hochschule Amberg-Weiden

Decaying ϵ-greedy exploration

- Decrease ϵ of ϵ-greedy exploration over time according to a special schedule
- Recap: ϵ-greedy exploration

$$\pi(a|s) = \begin{cases} 1 - \epsilon_t + \dfrac{\epsilon_t}{|\mathcal{A}(s)|} & \text{if } a \text{ maximizes } Q(s,a). \\ \dfrac{\epsilon_t}{|\mathcal{A}(s)|} & \text{for all other actions.} \end{cases}$$

- Decaying ϵ-greedy exploration: Update ϵ according to

$$\epsilon_t = \min\left(1, \frac{c}{d^2 t}\right)$$

c: tunable parameter > 0

t: time step

d: minimum gap $\min_a \Delta_a$

- Logarithmic asymptotic total regret but useless in practice, because gaps Δ_a are not known

Exploration and exploitation

Ostbayerische Technische Hochschule Amberg-Weiden

UCB1 (upper confidence bound)

- Estimate an upper confidence for each action value measuring the uncertainty of the true Q-function
- Depends on the number $N_t(a)$ than an action has been selected until time t
  - small $N_t(a)$ → estimated value is uncertain
  - large $N_t(a)$ → estimated value is certain
- UCB1 algorithm: Select action maximizing upper confidence bound as

$$a_t = \underset{a \in \mathcal{A}}{\operatorname{argmax}} \left(Q(a) + \sqrt{\frac{2 \log t}{N_t(a)}} \right)$$

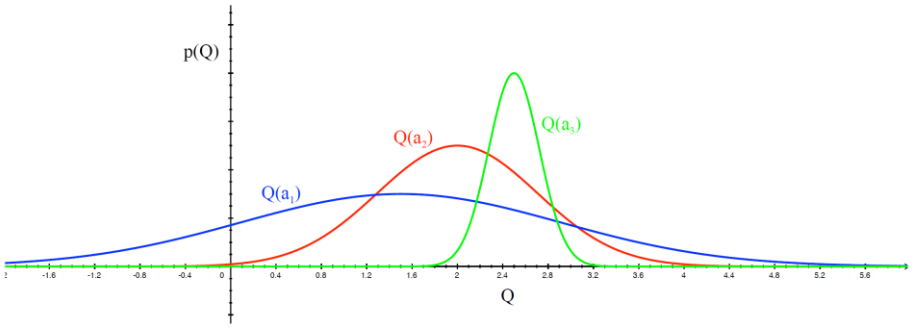
- Logarithmic asymptotic total regret (but larger than lower bound), easy to implement
- Standard for a lot of practical problems

Exploration and exploitation

Ostbayerische Technische Hochschule Amberg-Weiden

Thompson Sampling

- "randomly take action according to the probability you believe it is the optimal action" (Thompson 1933)
- Use internal (probabilistic) model to approximate the true Q-function
- At every time step
  - Sample from the approximated Q-function for every action
  - Select action with best sample outcome
  - Observe reward, update Q-function approximation
- Logarithmic asymptotic total regret (achieving lower bound)



https://www.deepmind.com/learning-resources/introduction-to-reinforcement-learning-with-david-silver

Exploration and exploitation

Ostbayerische Technische Hochschule Amberg-Weiden

Example: Thompson sampling

- 2 Bernoulli bandits (reward of 1 with prob. μ_i, otherwise 0)
- $\mu = [0.6, 0.4]$
- Rewards obtained so far shown on right side

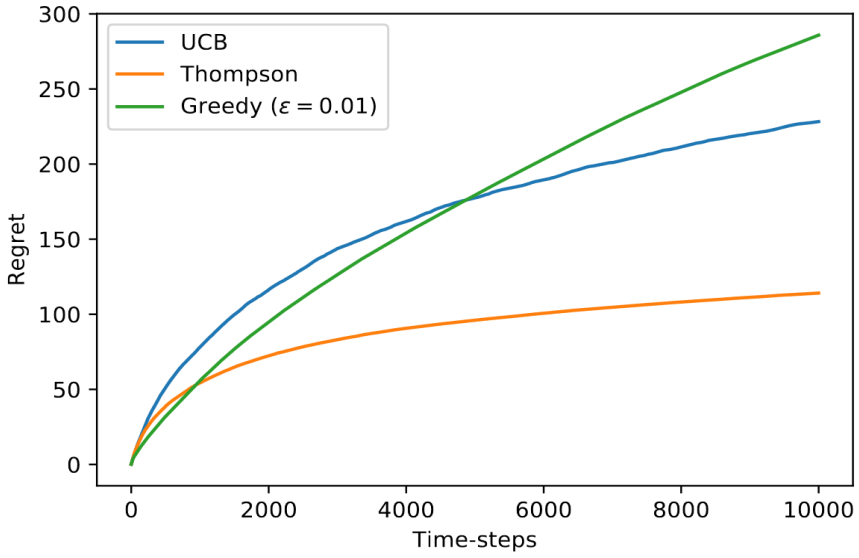
- At time step $t = 14$
  - Approximate both bandits with e.g. beta-distribution
    Bandit 1: $\alpha = 4, \quad \beta = 2$
    Bandit 2: $\alpha = 3, \quad \beta = 4$
  - Sample from approximated Q-function for both bandits, e.g.
    Bandit 1: $Q_{sampled} = 0.68$
    Bandit 2: $Q_{sampled} = 0.56$
  - Select action with best sample outcome → Bandit 1
  - Observe reward of 1, update Q-function approximation
    Bandit 1: $\alpha = 5, \quad \beta = 2$
    Bandit 2: $\alpha = 3, \quad \beta = 4$

time	bandit	result
1	1	1
2	1	0
3	2	1
4	2	1
5	1	1
6	2	1
7	2	1
8	2	0
9	1	1
10	2	0
11	1	0
12	2	0
13	1	1

Exploration and exploitation

Example

- 5 Bernoulli bandits (give reward of 1 with probability μ_i, otherwise 0)
- $\mu = [0.6, 0.5, 0.4, 0.3, 0.2]$
- Shown is total regret over number of pulls



https://team.inria.fr/polaris/files/2019/05/Gast.pdf

Exploration and exploitation

Ostbayerische Technische Hochschule
Amberg-Weiden

Example: Comparison of ϵ-greedy / UCB / Thompson

Example: Comparison of ϵ-greedy / UCB / Thompson

Example: Comparison of ϵ-greedy / UCB

Exploration and exploitation

Ostbayerische Technische Hochschule Amberg-Weiden

Comparison of all presented methods

Method	Pro	Con
Optimistic initialization	Simple, works well in practice	Linear total regret
R-max	Simple, works well in practice?	Linear total regret
ϵ-greedy	Simple, works well in practice	Linear total regret, param. ϵ
decaying ϵ-greedy	Near optimal	Unfeasible in practice
UCB1	Simple, works ok, near optimal	
Thompson sampling	Optimal	Complex for generic MDPs

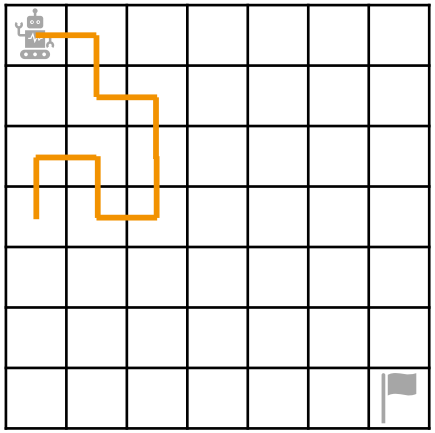
Exploration and exploitation

Ostbayerische Technische Hochschule
Amberg-Weiden

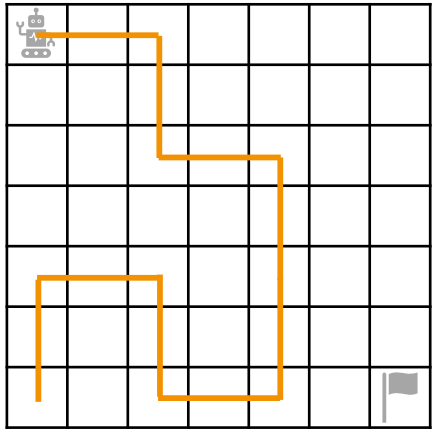
Other exploration techniques

Multi-step actions

- Execute the same action for n steps
- Well suited for continuous environments



1-step action



2-step action

Exploration and exploitation

Ostbayerische Technische Hochschule
Amberg-Weiden

Count-based exploration

- Informal: Count how often you have been in each state and try to explore states that you haven't been to many times yet
- Use additive reward term of the form

$$R = \alpha R_{ext} + \beta R_{int}$$

  to favor exploration of new states (with weighting factors α, β)
- R_{ext} (extrinsic reward) is the original reward from the RL task (extrinsic reward)
- R_{int} (intrinsic reward) is a count-based exploration reward
  (high if state hasn't been visited often)
- Easy to implement for discrete states (need to store how often each state has been visited) but can be generalized also to continuous states

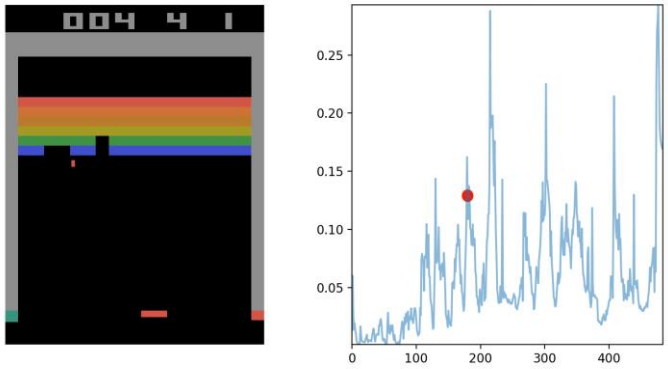
Exploration and exploitation

Ostbayerische Technische Hochschule Amberg-Weiden

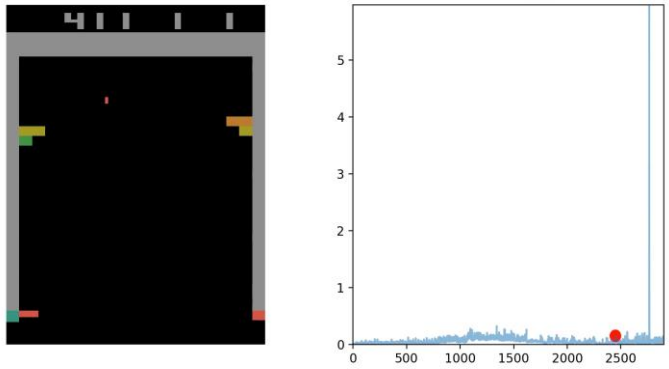
Prediction-based rewards

- Train a model to predict the outcome of the next action
- If the predicted outcome differs heavily from the true outcome (due to new states/rewards never seen before), assign a high intrinsic R_{int} reward to the corresponding action (similar to count-based exploration)

intrinsic reward at the beginning



intrinsic reward when passing the level for the first time



https://openai.com/blog/reinforcement-learning-with-prediction-based-rewards/

Note: Both prediction- and count-based rewards belong to the class of curiosity driven exploration

Exploration and exploitation

Ostbayerische Technische Hochschule Amberg-Weiden

Auxiliary tasks

- Fancy variant of reward shaping: Consider additional tasks (with corresponding rewards) that the agent has to solve along its way of solving the main task

Example

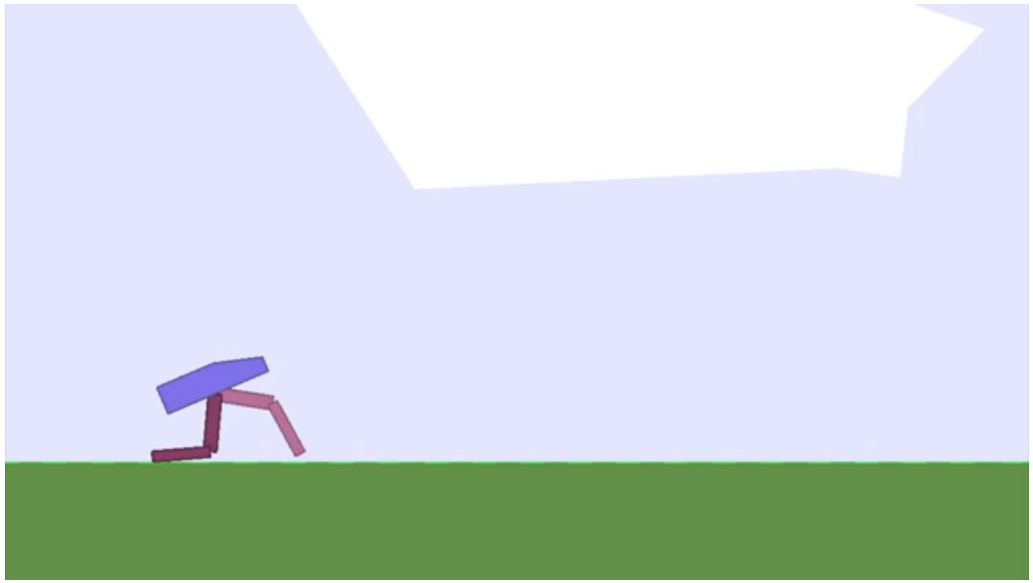
- Main task: Robot has to open a box, place an object into it and close the lid of the box afterwards again
- Auxiliary task 1: Box has been opened
- Auxiliary task 2: Object has been placed in the box
- Auxiliary task 3: Box has been closed

Exploration and exploitation

Ostbayerische Technische Hochschule
Amberg-Weiden

Curriculum learning

- Problem: Sometimes exploration does not help at all because the task is too difficult to be solved by an untrained agent
- Solution: Create a curriculum of environments, i.e. environments with increasing complexity until the previously unsolvable task can be solved



https://www.youtube.com/watch?v=D1WWhQY9N4g

Exploration and exploitation

Ostbayerische Technische Hochschule
Amberg-Weiden

Brief summary

- Exploration vs. exploitation (sticking to known stuff vs. exploring something new) is a predominant problem in reinforcement learning
- There exist multiple methods with different implementation complexity favoring exploration
- A good overview of different methods is provided here

Kahoot!

Ostbayerische Technische Hochschule
Amberg-Weiden