An Introduction to Reinforcement Learning

Hossein Hajiabolhassan

Department für Mathematik
und Informationstechnologie
Montanuniversität Leoben

Zhejiang Normal University Seminar

November 11, 2022

Table of Contents

MONTAN UNIVERSITÄT LEOBEN

Preliminary

Reinforcement Learning

Definition (Reinforcement Learning)

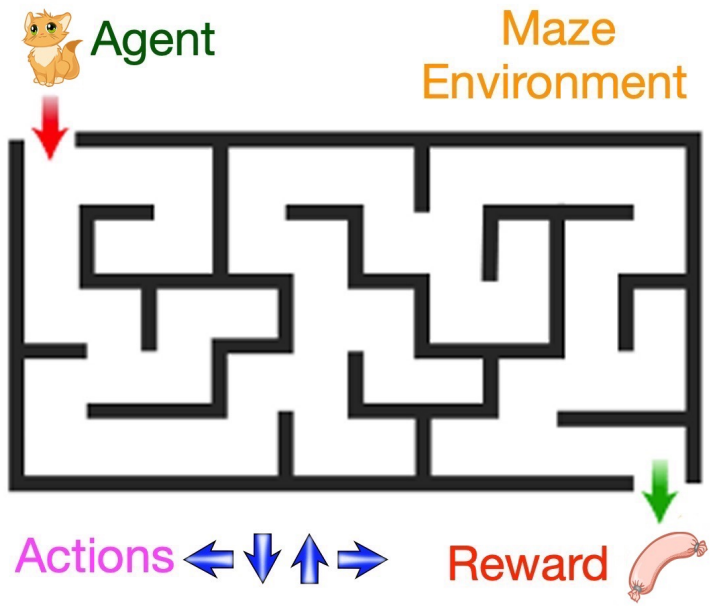
Training: Reinforcement Learning is the science of decision making. It is about learning the optimal behavior in an environment to obtain maximum reward and minimum punishment.
Task: Find a model to maximize the total cumulative reward of the agent.
Examples: Teach an artificial intelligence (AI) system to play GO

▸ Reference

MONTAN UNIVERSITÄT LEOBEN

Training

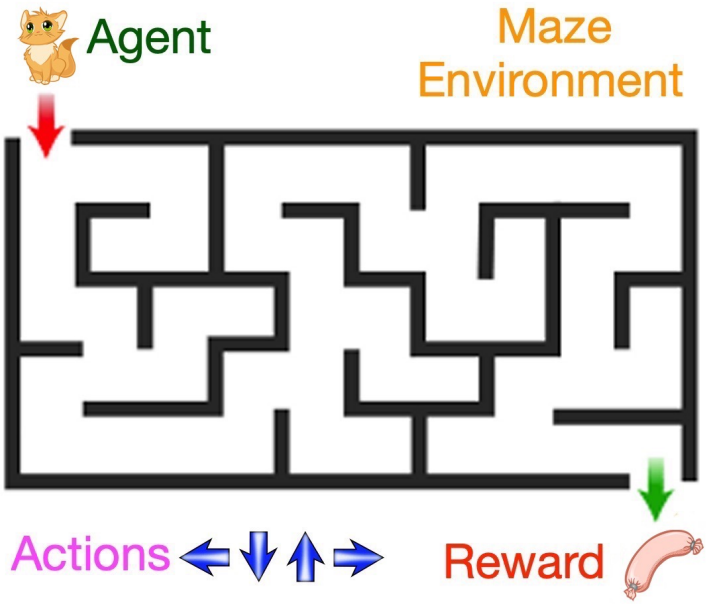
How is it possible to train a cat to find exit easily?



References 1, 2, 3

Reinforcement Learning

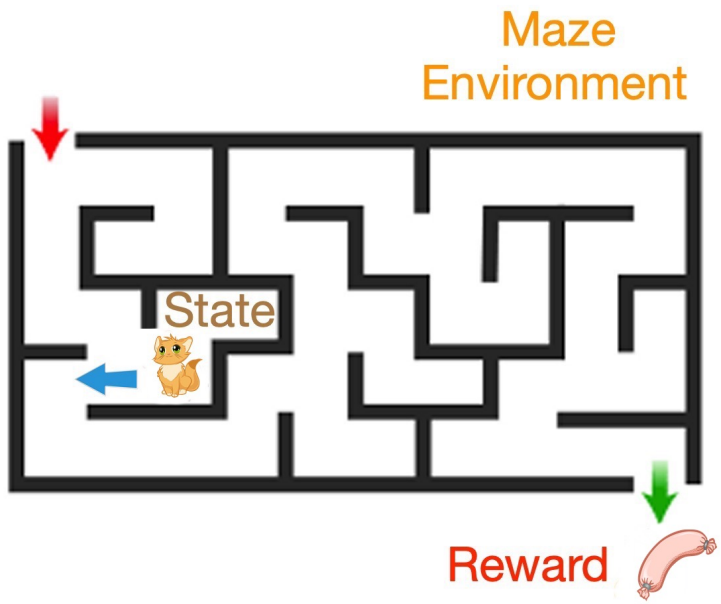
- Agent: The learner or decision-maker.
- Environment: The Agent's world in which it lives and interacts.
- The goal of reinforcement learning is to train an agent to complete a task within environment



▸ References 1, ▸ 2, ▸ 3

Reinforcement Learning

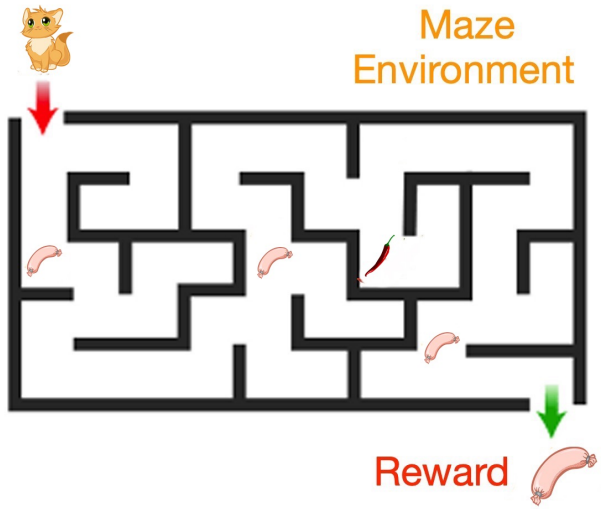
- Action: The set of all possible moves an agent can make. The agent can interact with the environment by performing some action but cannot influence the rules or dynamics of it by those actions.
- State: An immediate situation in which the agent finds itself. It can be a specific moment or position in the environment.



References 1, 2, 3

Reinforcement Learning

- Reward: For every action made, the agent receives a reward from the environment and indicates performance of the agent.
- Reward Hypothesis: All goals can be described by the optimization of some function of the sequence of rewards.
- Agent is not so clever to learn alone. Agent's goal is to optimize some function of the sequence of rewards to reach the goal.



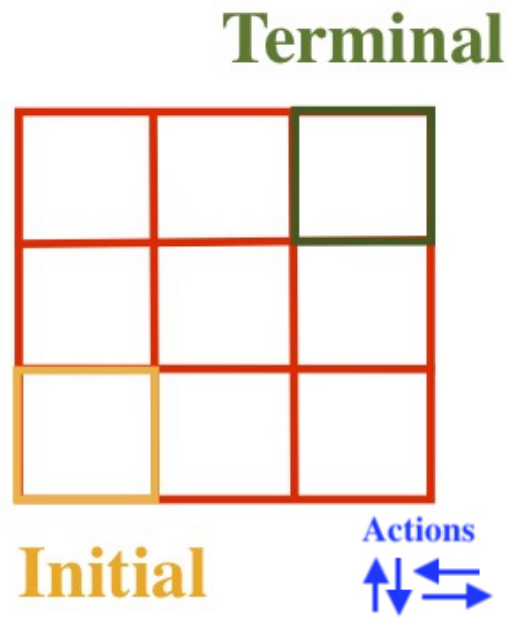
▸ References 1, ▸ 2, ▸ 3, ▸ 4

MONTAN UNIVERSITÄT LEOBEN

Training an Agent: Assigning Rewards

Goal

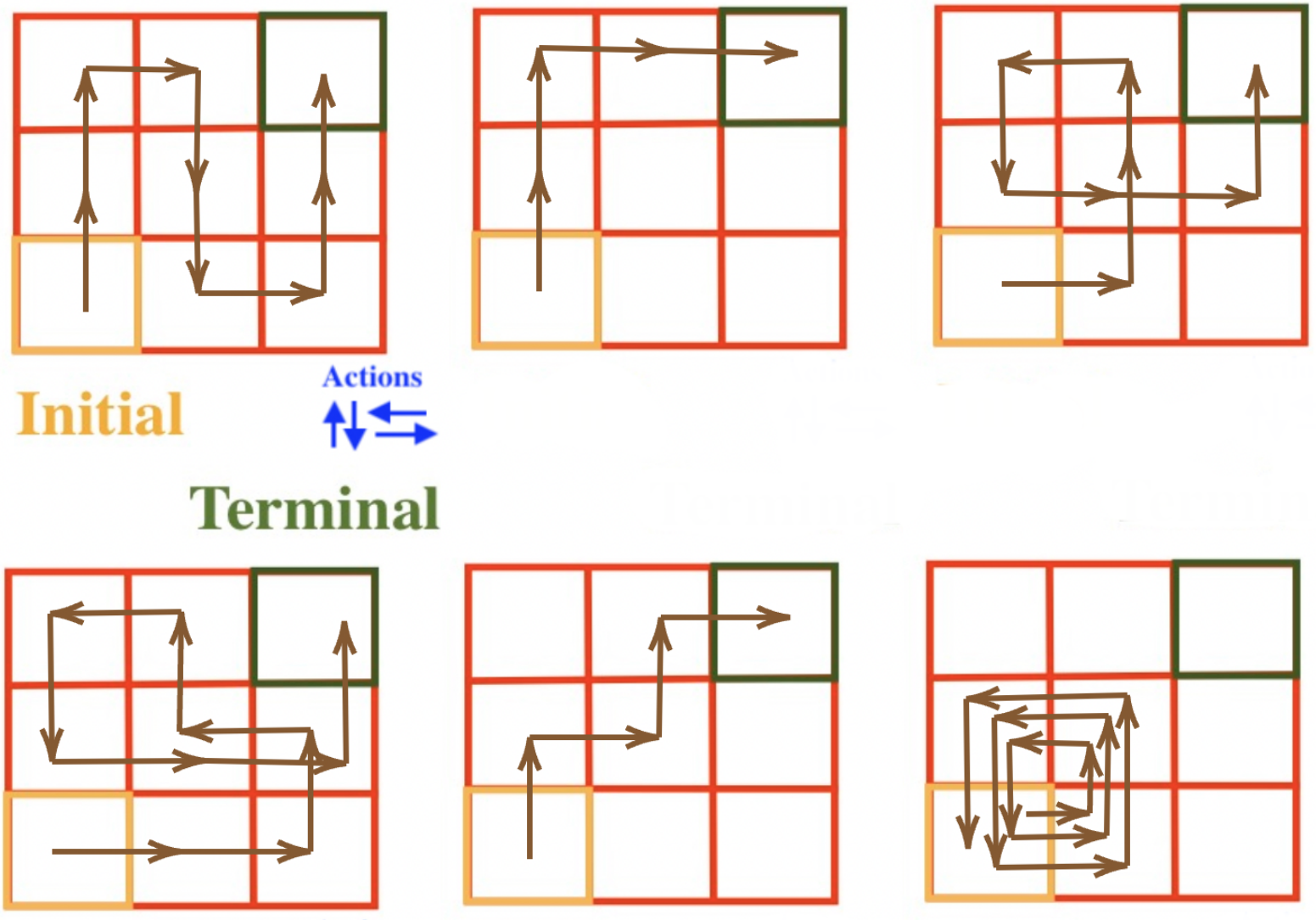
- Shortest Path: Train an agent which is in the initial state to reach to the terminal state in the shortest possible time.
- How is it possible to train the agent to reach the goal?



1. Terminal State is an absorbing state. All actions taken in the absorbing state lead back to that same state.
2. Agent learns by doing actions and interacting with environment (collecting data: trial and error, former experiences, ...)
3. By assigning appropriate rewards, an agent must be able to learn from its own experiences.

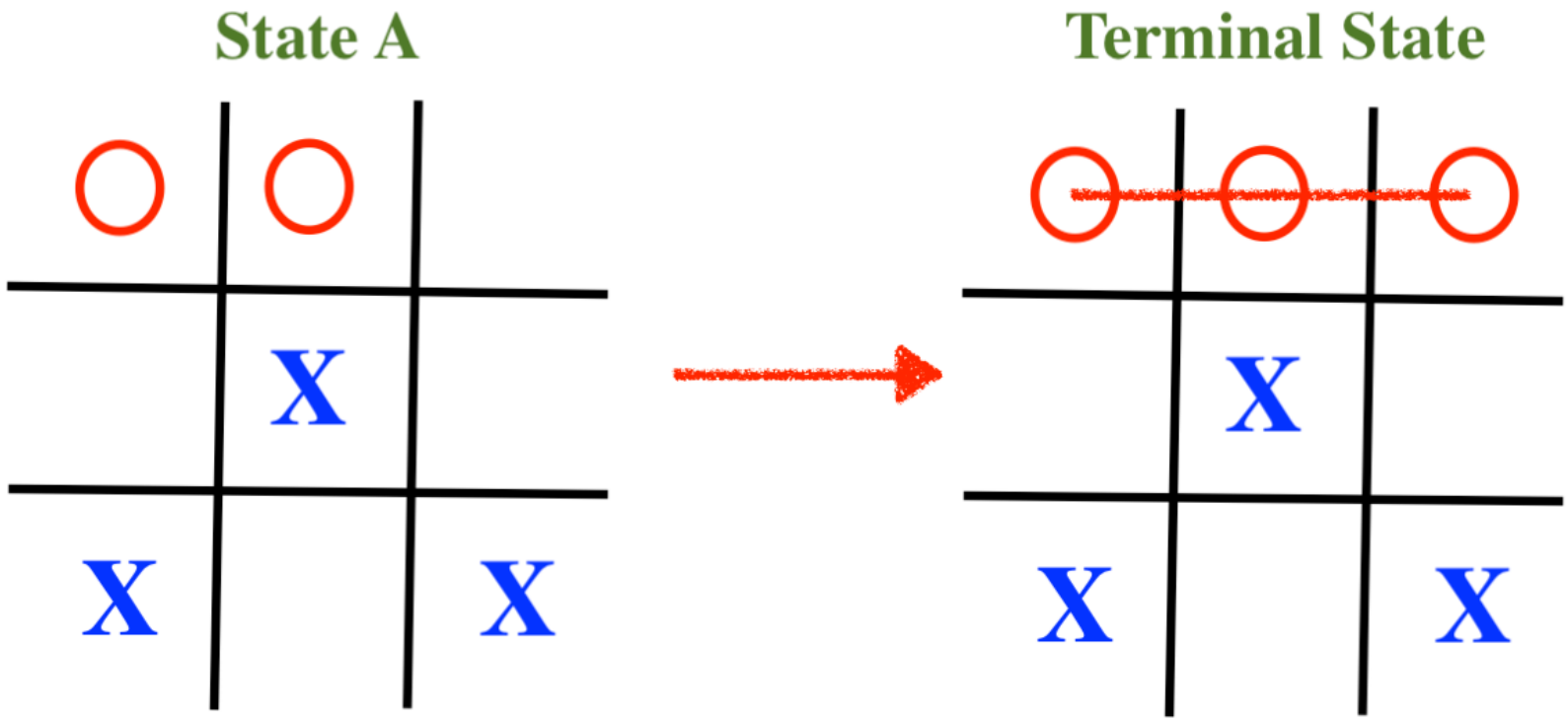
MONTAN UNIVERSITÄT LEOBEN

Experiences



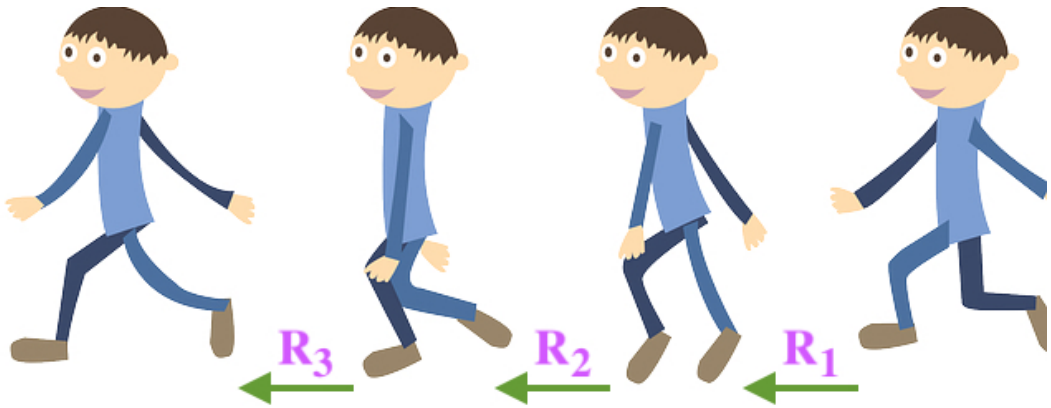
Experiences

For instance in Tic-Tac-Toe, collecting data does not mean to have all of wining strategies! Also, learning does not mean to copy a wining strategy!

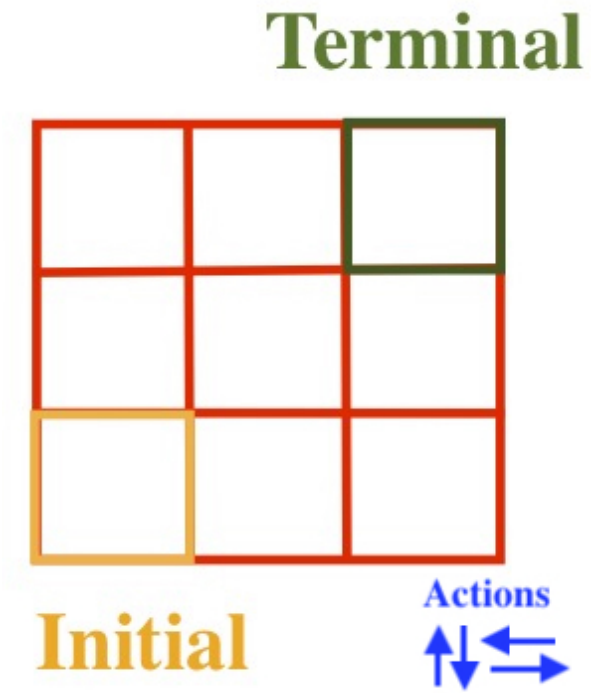


Choosing a Reward Function

- We usually consider the cumulative sum of rewards as a function to reach the goal (Cumulative Sum: $R_1 + R_2 + \cdots + R_n$). In this case, the existence of the terminal state is necessary.



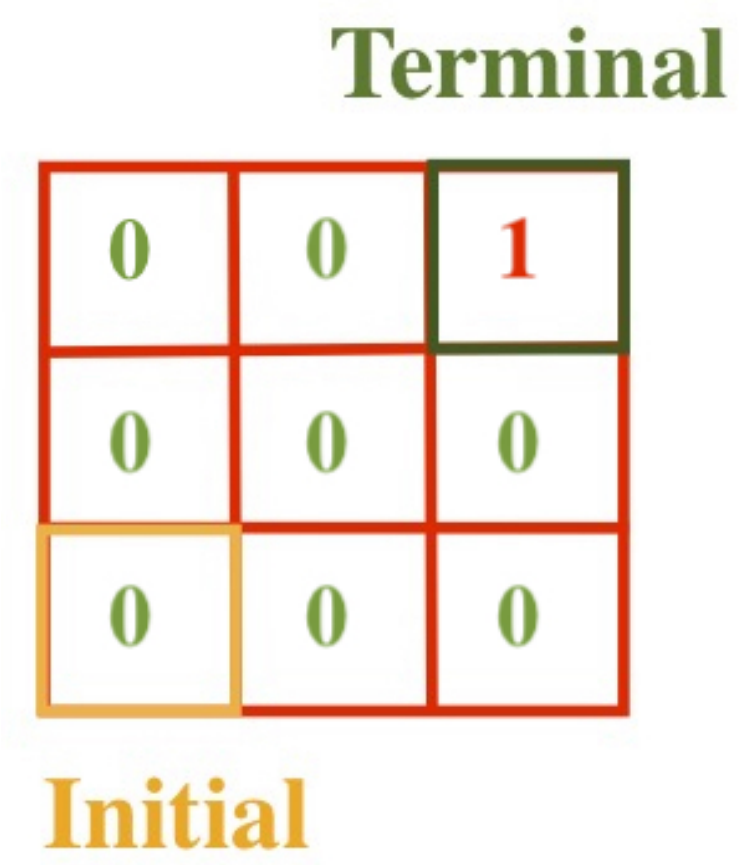
▸ Reference



Choosing a Reward Function

What is the best candidate?

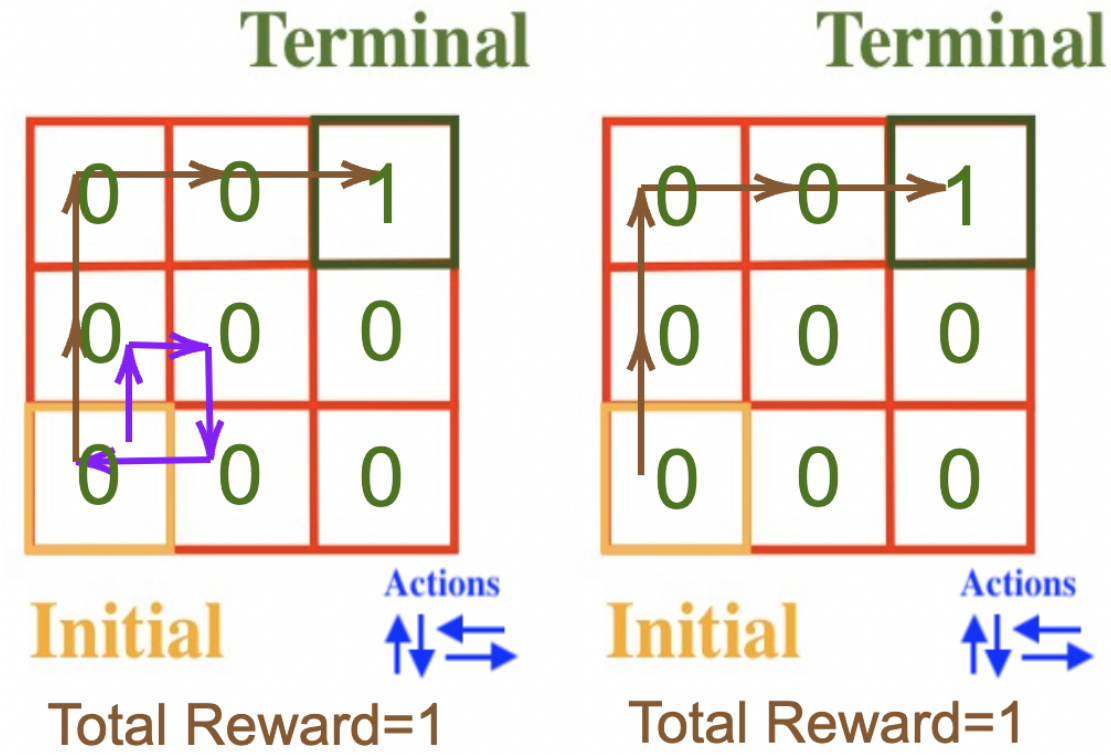
- Cumulative Sum: $R_1 + R_2 + \cdots + R_n$
- All reward on JUST terminal state (goal)!



Choosing a Reward Function

What is the best candidate?

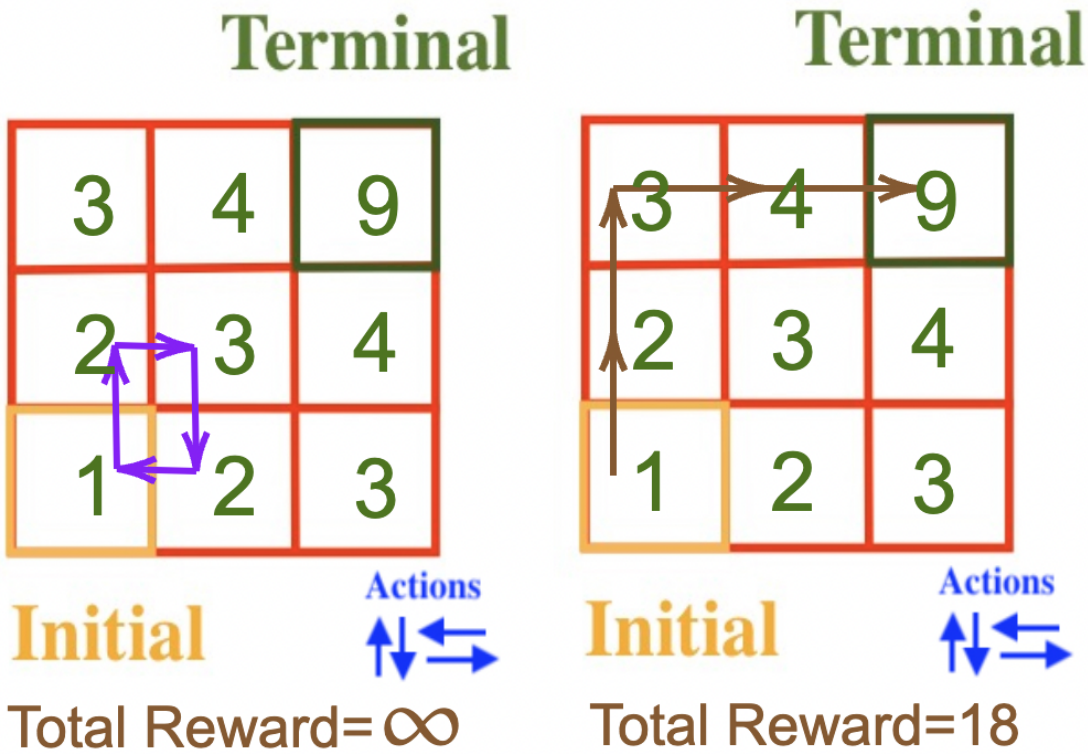
- Cumulative Sum: $R_1 + R_2 + \cdots + R_n$
- All reward on JUST terminal state (goal)!



Choosing a Reward Function

What is the best candidate?

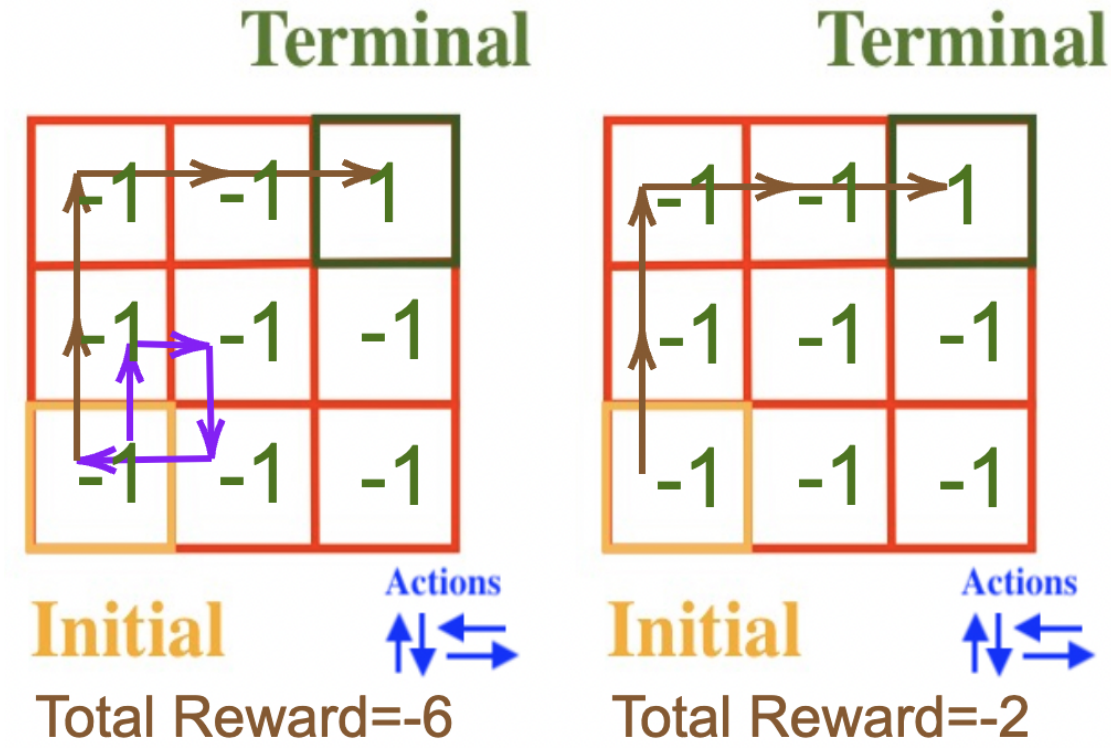
- Increasing reward as you approach terminal states.
- If there exists a cycle, agent can stay in the cycle forever or if the number of steps is large enough, then the summation will be a very large number!



Choosing a Reward Function

What is the best candidate?

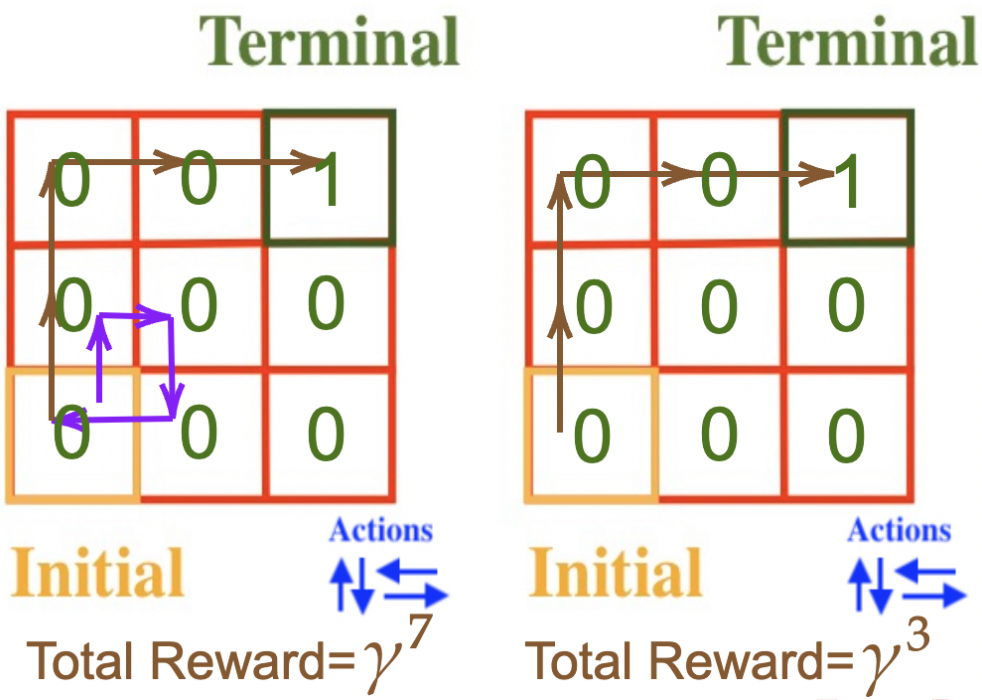
- What is the solution?
- Negative reward everywhere (except actions of terminal state)



Choosing a Reward Function

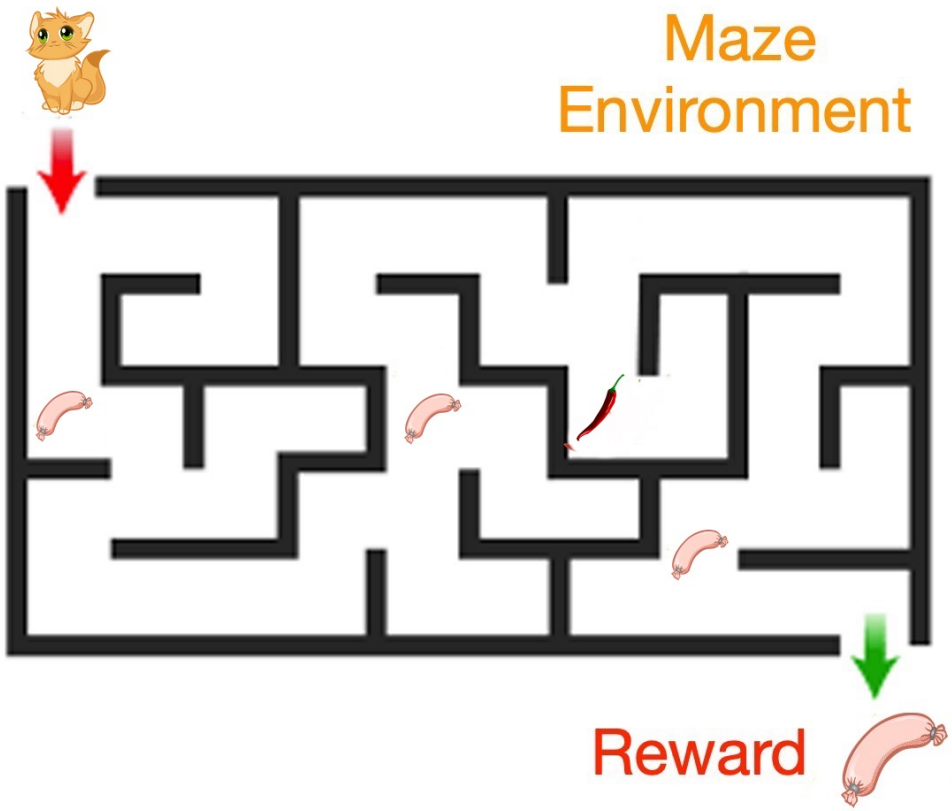
What is the best candidate?

- Valuing immediate money more than future money is a rational behavior known as discounting.
- We can consider the discounting rewards $\sum_{i=1}^{t} \gamma^{i-1} R_i$ where $0 < \gamma < 1$.



Algorithm

- We assign the rewards such that it is possible to train an agent. In training an AI, we should consider an appropriate algorithm.



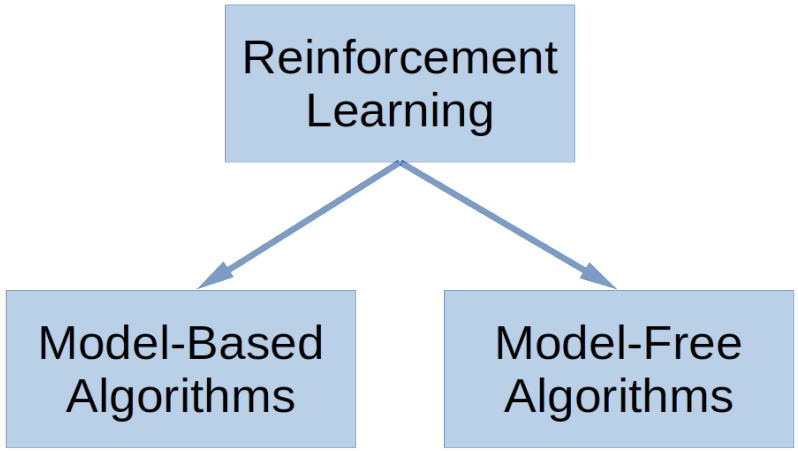
References 1, 2, 3, 4

Computational Strategies: Challenges and Limitations

Algorithms

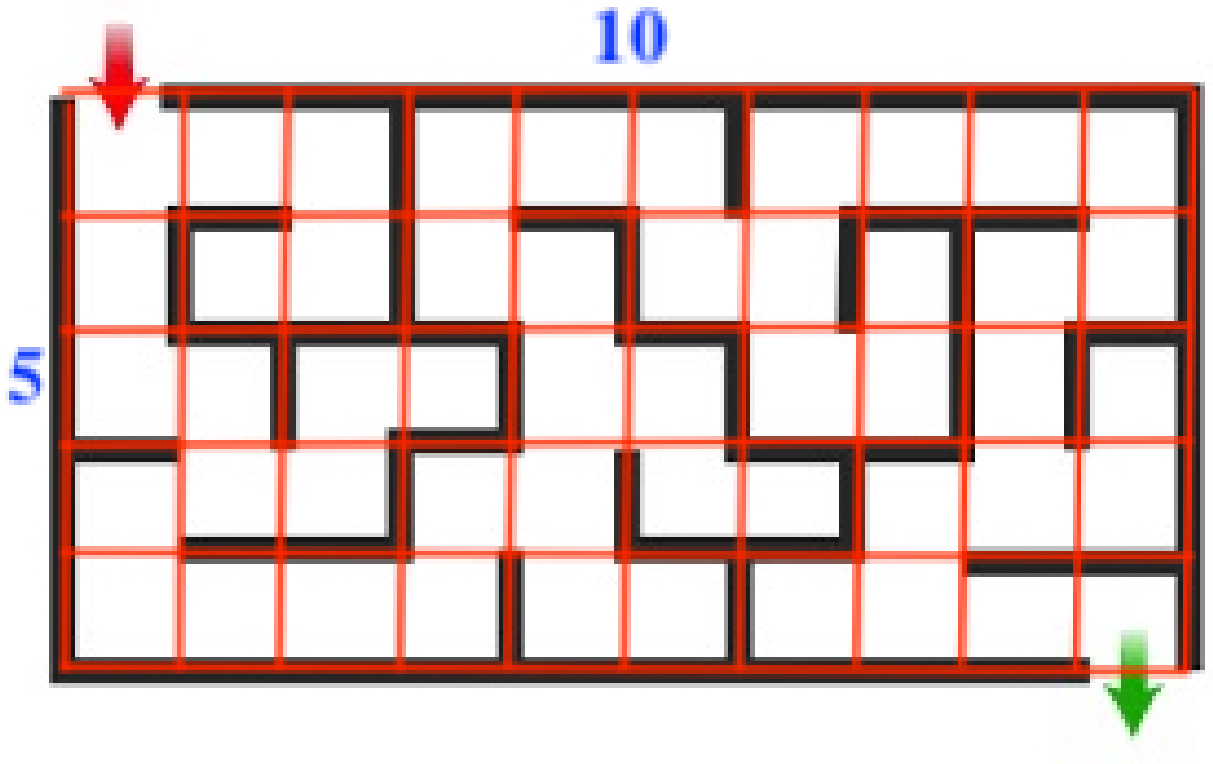
Models

1. Model-based algorithms use a predictive model of the environment to ask questions of the form "what will happen if I do action a?" to choose the best action a
2. Model-free algorithms seek to learn the consequences of their actions through experience



Known Environment

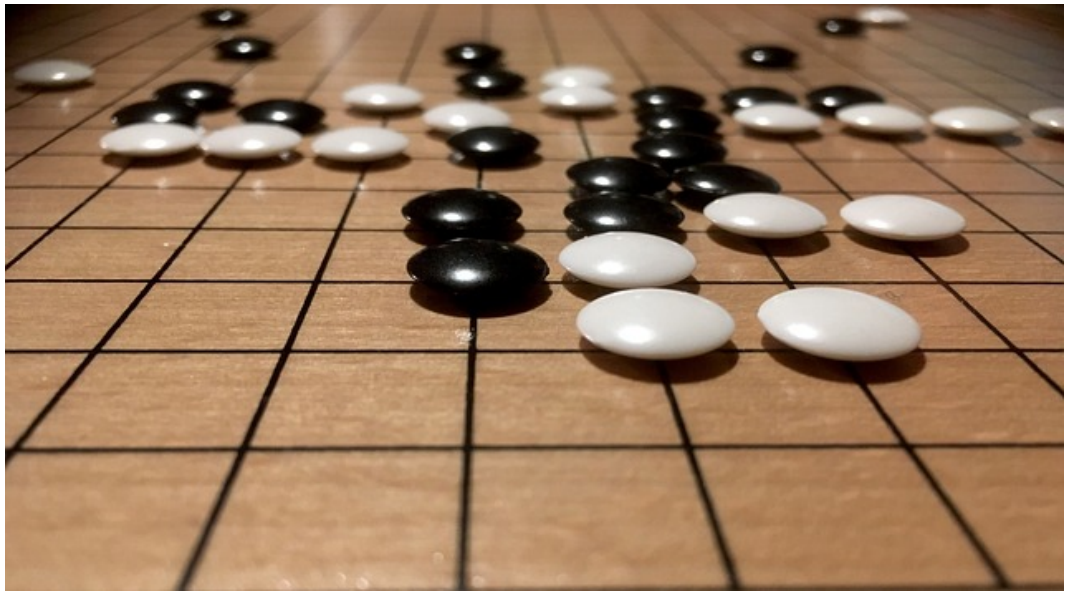
- Question: How many states has this environment?
- Challenge: The environment is usually large.



Reference

Known Environment

1. In GO, the number of possible board positions is around 10^{172}.
2. There are between 10^{78} to 10^{82} atoms in the observable universe.



▸ Reference

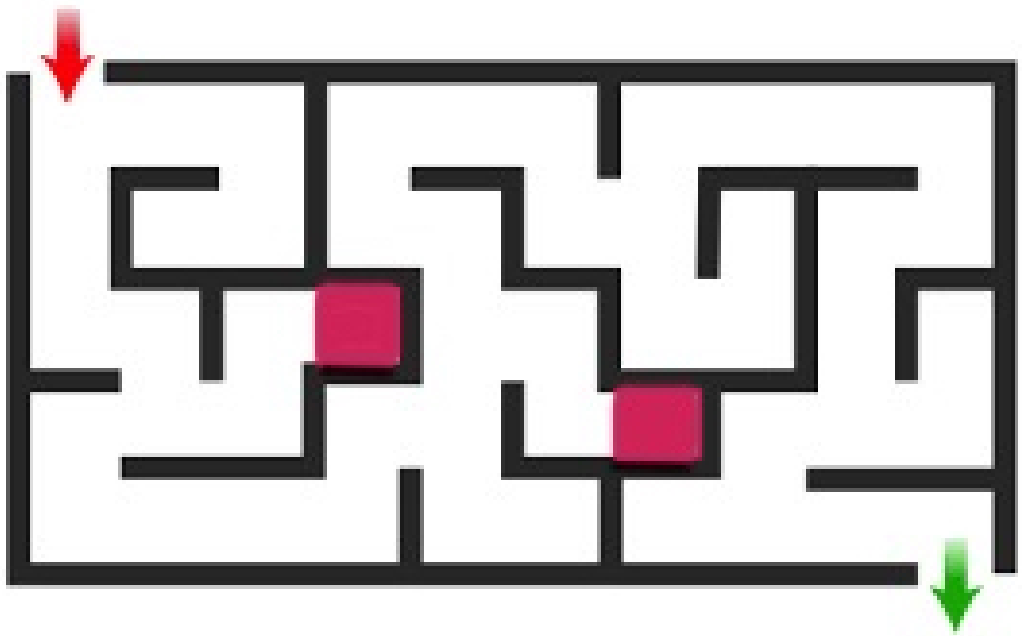
Unknown Environment

- At the beginning of exploring of the environment, if the agent do an action, then the agent moves to an unknown state.
- For instance, when we are playing a game and we should choose a door to continue the game and we are not awrae about next state.

▸ Reference

Unknown Environment

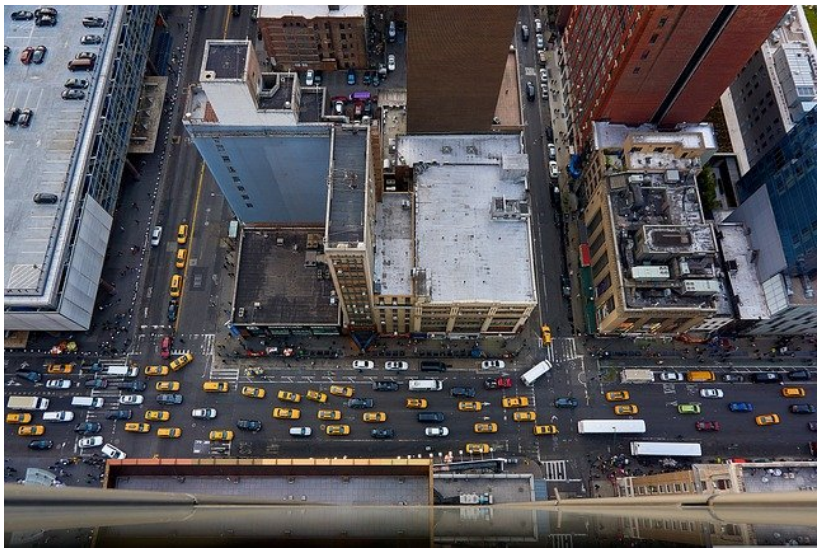
- Is it possible for the agent to distinguish between the red states? If we assume that all of walls are the same and high enough respect to the agent (the agent does not see the objects behind the wall).

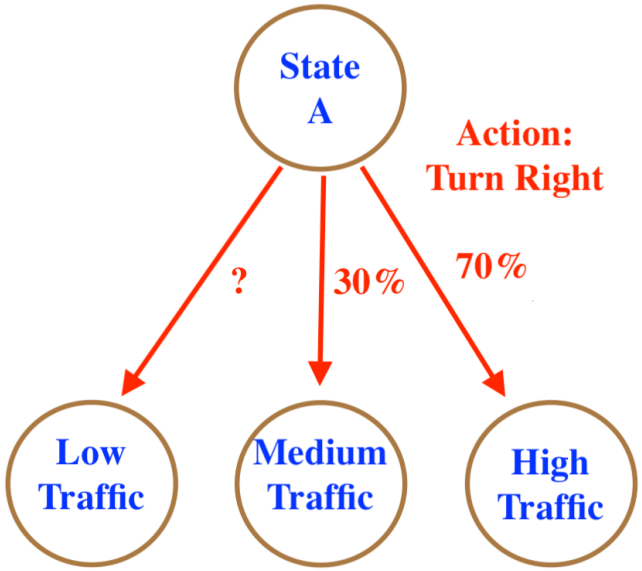


▸ Reference

Unknown Environment

- Actions may be stochastic and one may be unlucky to explore all of transitions. For instance, in the following Environment and in the state A, the action Turn Right is stochastic and the state Low Traffic has not been explored as its neighbour so far!

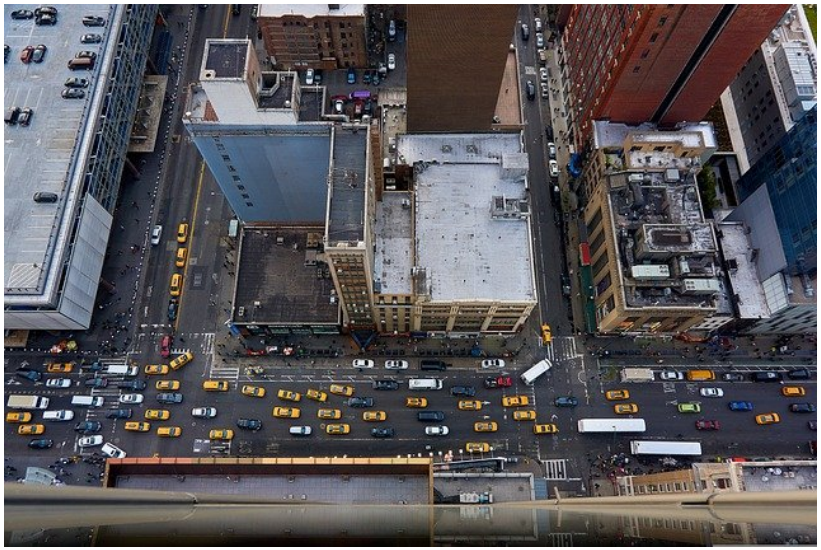


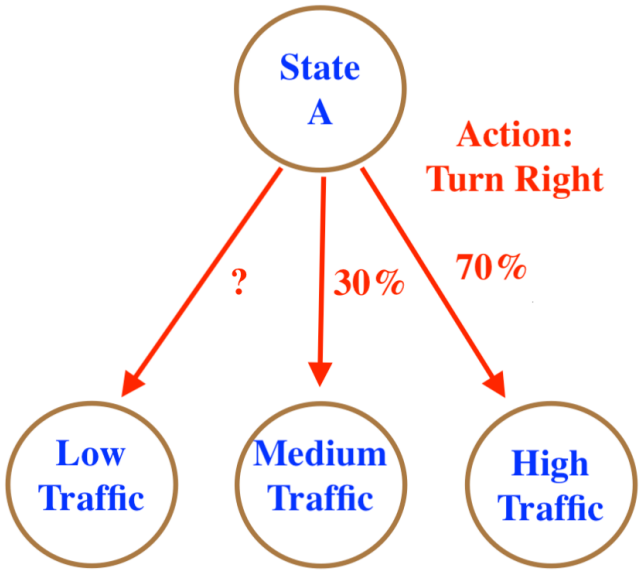


▸ Reference

Unknown Environment

- Limitation: The agent usually has no chance to run several actions sequentially in a specific state to explore the environment.
- Limitation: Agent usually cannot restart to explore the environment form an arbitrary state.





▸ Reference

Full Observability

- Challenge: Partial Observability

  In this talk, we assume full observability: the new state resulting from executing an action will be known to the system.

  An example for partial observability: in some card games (e.g. Bridge), the agent cannot see the hands of the opponent for the most part of the game.

▸ References

The Smallest Environment: Multi-Armed Bandits

Multi-Armed Bandit

Question: Is there any challenge for small environments?

Multi-Armed Bandit

Question: Is there any challenge for small environments?

"Bandit" is someone who robs people, especially one of a group of people who attack travellers.

▸ Reference

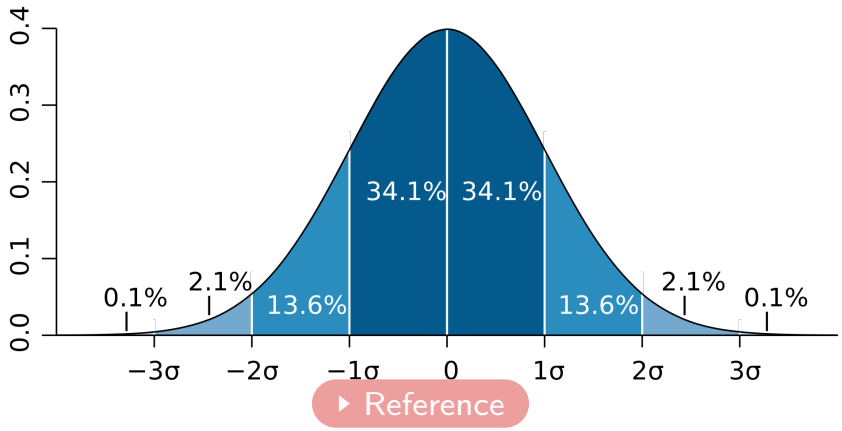
"Bandit" in "Multi-Armed Bandits" comes from "one-armed bandit" machines used in a casino.

▸ Reference

MONTAN UNIVERSITÄT LEOBEN

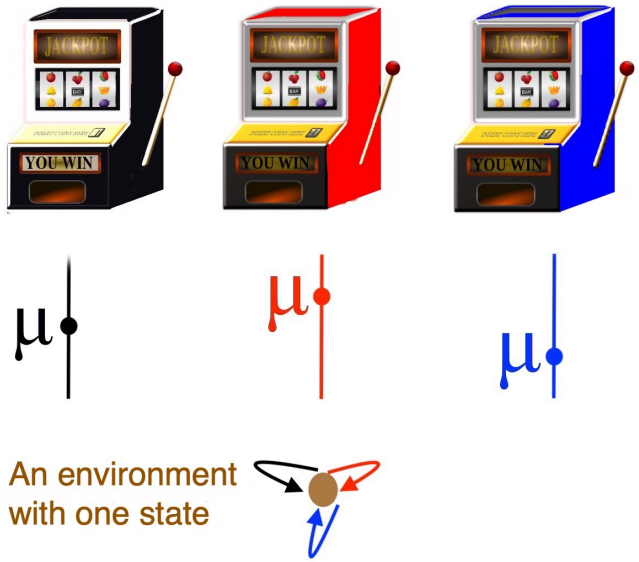
Rewards

- Question: Does a teacher always give the same reward when a student solve a kind of problem?
- The rewards and punishments are often non-deterministic, and there are invariably stochastic elements governing the underlying situation.
- Question: If the reward has normal distribution with unknown mean, how do we find a approximation of the mean of distribution?
- The law of large numbers states that as a sample size grows, its mean gets closer to the average of the whole population.



Small Environment

- Given a bandits with K arms where each arm has a fixed but unknown probability distributions with mean μ. Pulling any arm, gives you a stochastic reward. In this talk, we usually consider Gaussian Bandits.
- Gaussian Bandits: The rewards come from a normal distribution.
- Bernoulli Bandits: At each round, we receive a binary reward $(0-1)$.



▸ Reference

Goals in Multi-Armed Bandit

- Our Goal can be one of the following items:
  1. Pull the arms one-by-one in sequence such that we maximize our total reward collected in the long run.
  2. Best arm identification: minimize the error probability at time T.
  3. Best arm identification: minimize the total number of stages used to return the best arm with probability $1 - \delta$.

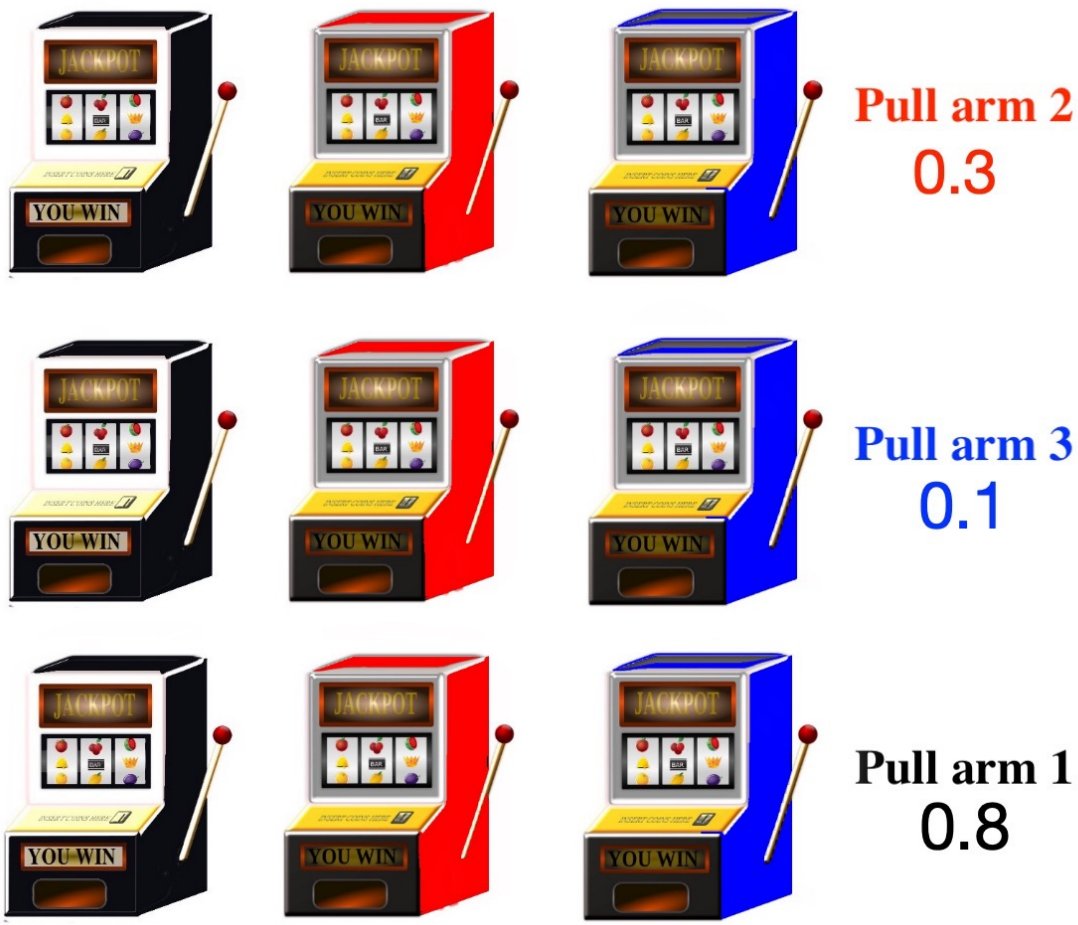
Training to Reach the Goal

One can use the rewards of bandits to train the agent.

▸ Reference

MONTAN UNIVERSITÄT LEOBEN

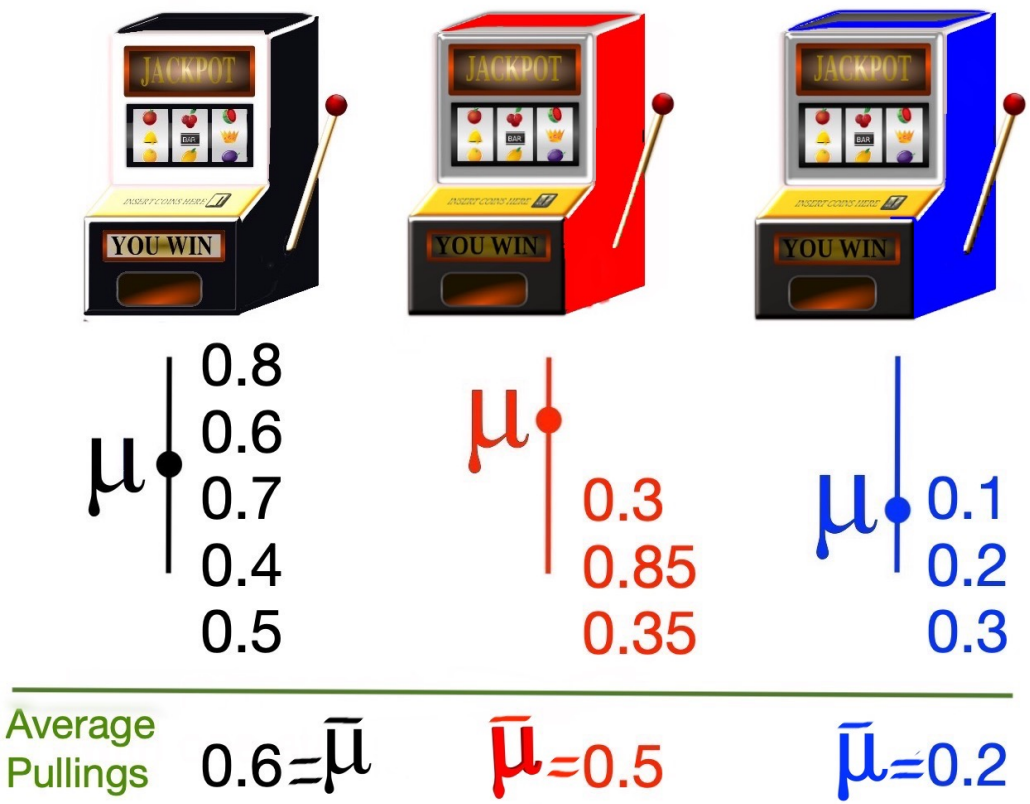
Gaussian Multi-Armed Bandit



▸ Reference

Gaussian Multi-Armed Bandit

- The multi-armed bandit problem



▸ Reference

Various Algorithms for Multi-Armed Bandits

Algorithm and Policy

The agent needs an algorithm to reach his/her goal!

- Goal: maximize the cumulative sum of rewards

Greedy Algorithm: Based on Exploitation

1. Pull all arms once! (Exploration)
2. Choose the best arm!
3. Pull the best arm again (Exploitation)
4. Recompute the average reward of arms
5. Go to step 2 and continue

References 1, 2, 3

- What is the best algorithm?

MONTAN UNIVERSITÄT LEOBEN

Performance Measure of Algorithms

1. There are K-arms.
2. $\mu^* = \max_{1 \leq i \leq K} \mu_i$ denotes the optimal mean of arms.
3. In each time step t, the agent has to play an arm $I_t \in \{1, 2, \ldots, K\}$.
4. Let $x_{I_t}(t)$ be the obtained reward at time t where $1 \leq t \leq T$.
5. The aims is to maximize the cumulative sum of rewards.
6. The rewards are stochastic, so we consider the expectation of them

$$\sum_{t=1}^{T} E(x_{I_t}(t)) = \sum_{t=1}^{T} \mu_{I_t}$$

7. Instead, one can minimize the pseudo regret

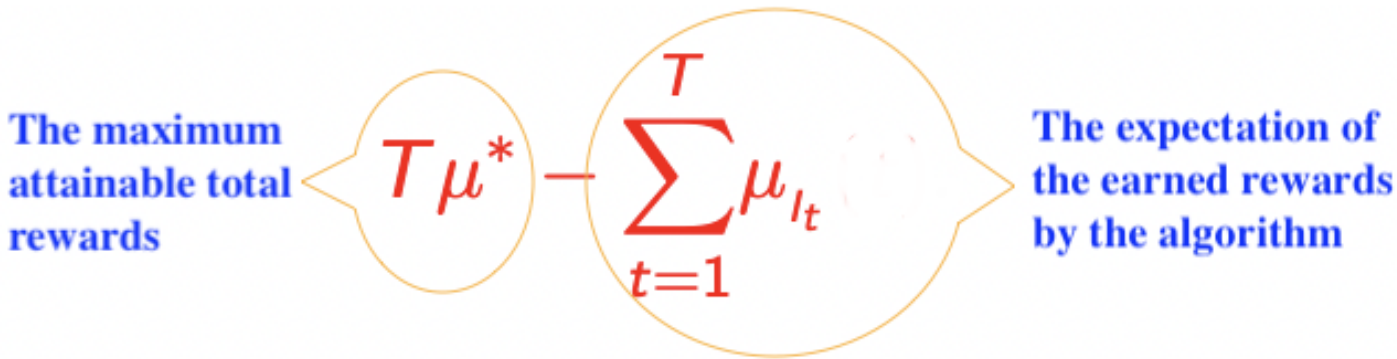
$$R(T) = T\mu^* - \sum_{t=1}^{T} \mu_{I_t}$$

MONTAN UNIVERSITÄT LEOBEN

Performance Measure of Algorithms

1. There are K-arms.
2. $\mu^* = \max_{1 \le i \le K} \mu_i$ denotes the optimal mean of arms.
3. In each time step t, the agent has to play an arm $I_t \in \{1, 2, \ldots, K\}$.
4. Let $x_{I_t}(t)$ be the obtained reward at time t where $1 \le t \le T$.
5. The aims is to maximize the cumulative sum of rewards.
6. The rewards are stochastic, so we consider the expectation of them

$$\sum_{t=1}^{T} E(x_{I_t}(t)) = \sum_{t=1}^{T} \mu_{I_t}$$



Algorithm and Policy

Greedy Algorithm: Explore-First with parameter N

1. Pull each of K arms N times! (Exploration)
2. Choose the best arm!
3. Pull the best arm again (Exploitation)
4. Recompute the average reward of arms
5. Go to step 2 and continue

Theorem

If T is known, Explore-first achieves regret:

$$R(T) \leq T^{\frac{2}{3}} \times O(K \log T)^{\frac{1}{3}}.$$

The performance of the exploration phase is terrible.

MONTAN UNIVERSITÄT LEOBEN

Lower Bound for Regret

Theorem (P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire)

Fix time horizon T and the number of arms K. For any bandit algorithm, there exists a problem instance such that

$$R(T) \geq \Omega(\sqrt{KT}).$$

Theorem

If T is known, Explore-first achieves regret:

$$R(T) \leq T^{\frac{2}{3}} \times O(K \log T)^{\frac{1}{3}}.$$

The performance of the exploration phase is terrible.

MONTAN UNIVERSITÄT LEOBEN

The Exploration-Exploitation Dilemma

Challenge

The Exploration-Exploitation Dilemma

Exploitation: make the best decision given current information
Exploration: gather more information

1. The best long-term strategy may involve short- term sacrifices
2. Gather enough information to make the best overall decision

▸ Reference

MONTAN UNIVERSITÄT LEOBEN

Food Selection in Food Court

Exploit: Eat your favourite food

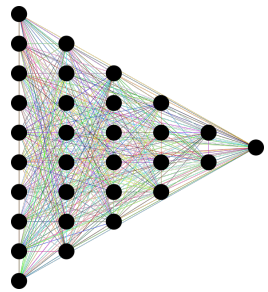
▸ Reference

Explore: Try a new food

▸ Reference

Examples of The Exploration-Exploitation Dilemma

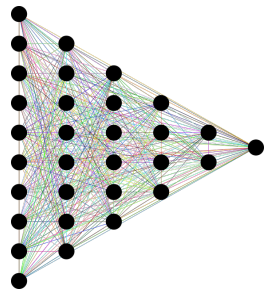
1. Hyperparameter Tuning in Machine Learning:
   Exploit: Use the best known Hyperparameters
   Explore: Check new Hyperparameters



▸ Reference

Examples of The Exploration-Exploitation Dilemma

1. Hyperparameter Tuning in Machine Learning:
   Exploit: Use the best known Hyperparameters
   Explore: Check new Hyperparameters



Reference

2. Skill Improvement:
   Exploit: Use current skills
   Explore: Learn new skills

Reference

3. Online Advertising:
   Exploit: Show the most successful advert
   Explore: Show a new advert

Different Algorithms: ϵ_T-Greedy

- Explore with probability $\epsilon_T = T^{-\frac{1}{3}}(K \log T)^{\frac{1}{3}}$
- Exploit with probability $1 - \epsilon_T$

1. Pull each of K arms once!
2. At time step T, pull the best arm with probability $1 - \epsilon_T$, otherwise pull an arbitrary arm with probability ϵ_T.
3. Recompute the average reward of arms and Go to step 2.

Question: In food selection, we usually use ϵ-Greedy algorithm. Which ϵ_T do you use?

▸ Reference

▸ Reference

MONTAN UNIVERSITÄT LEOBEN

Different Algorithms: ϵ_T-Greedy

- Explore with probability $\epsilon_T = T^{-\frac{1}{3}}(K \log T)^{\frac{1}{3}}$
- Exploit with probability $1 - \epsilon_T$

1. Pull each of K arms once!
2. At time step T, pull the best arm with probability $1 - \epsilon_T$, otherwise pull an arbitrary arm with probability ϵ_T.
3. Recompute the average reward of arms and Go to step 2.

Theorem

Algorithm ϵ_T-Greedy achieves regret:

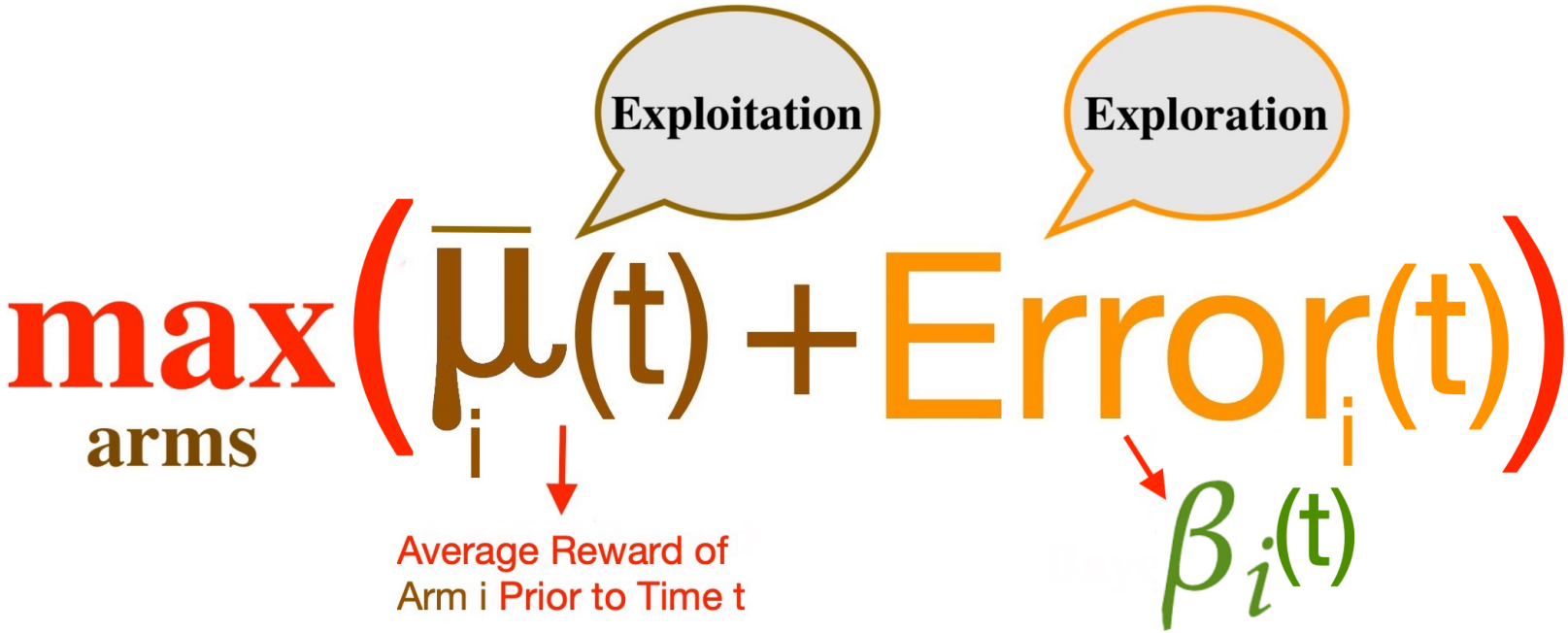
$$R(T) \leq T^{\frac{2}{3}} \times O(K \log T)^{\frac{1}{3}}.$$

The performance of the exploration phase is still terrible.

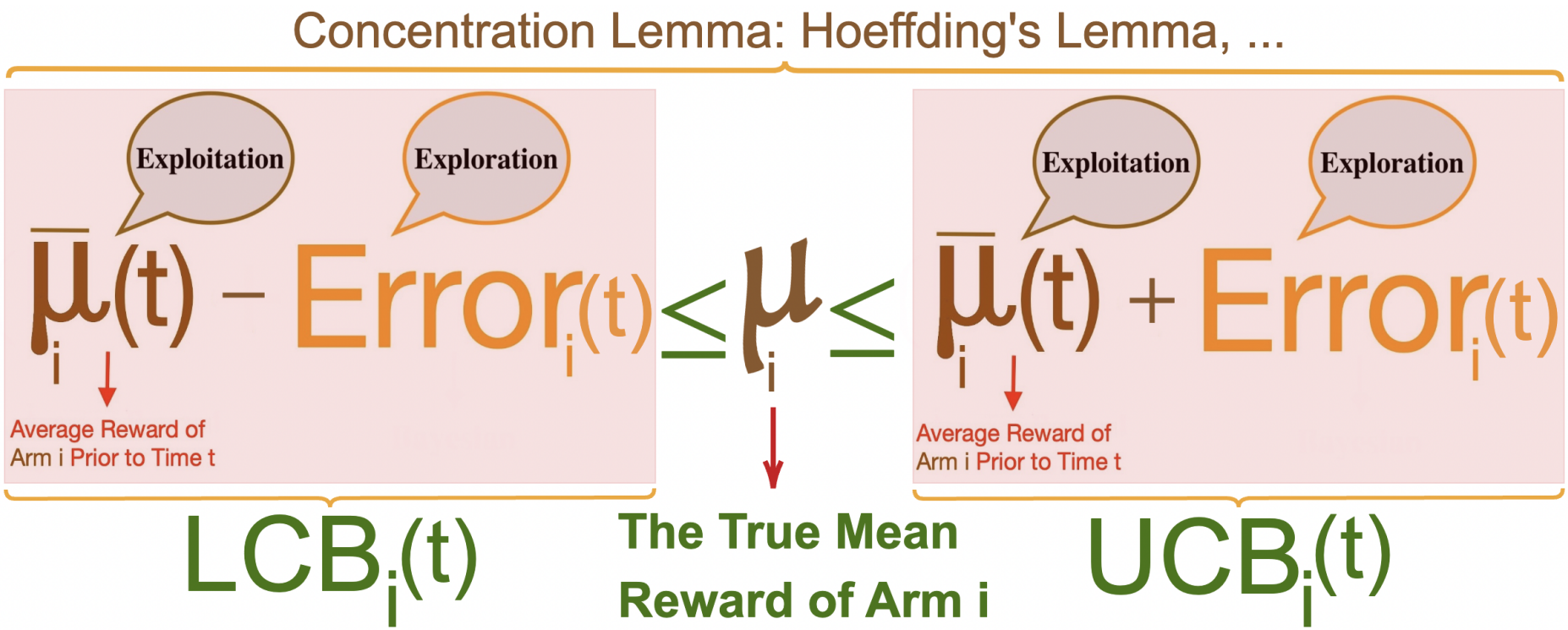
MONTAN UNIVERSITÄT LEOBEN

Advanced Algorithms (Adaptive Exploration)

1. Pull each of K arms once!
2. Pull the arm whose empirical average reward prior to time t plus its error has the maximum value and repeat this step!

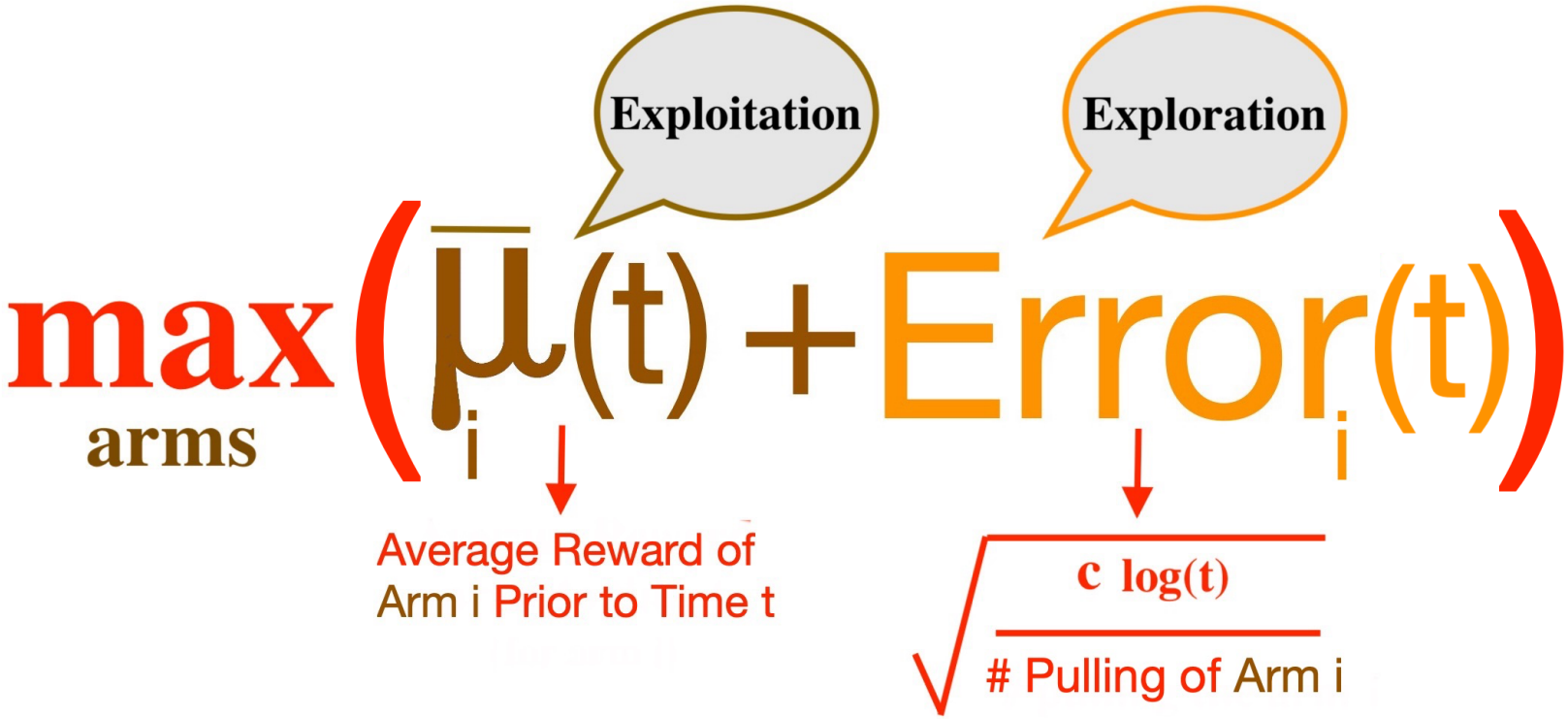


Advanced Algorithms (Adaptive Exploration)



Upper Confidence Bound Bandit (UCB1)

1. Pull each of K arms once!
2. Pull the arm whose empirical average reward prior to time t plus its error has the maximum value and repeat this step!



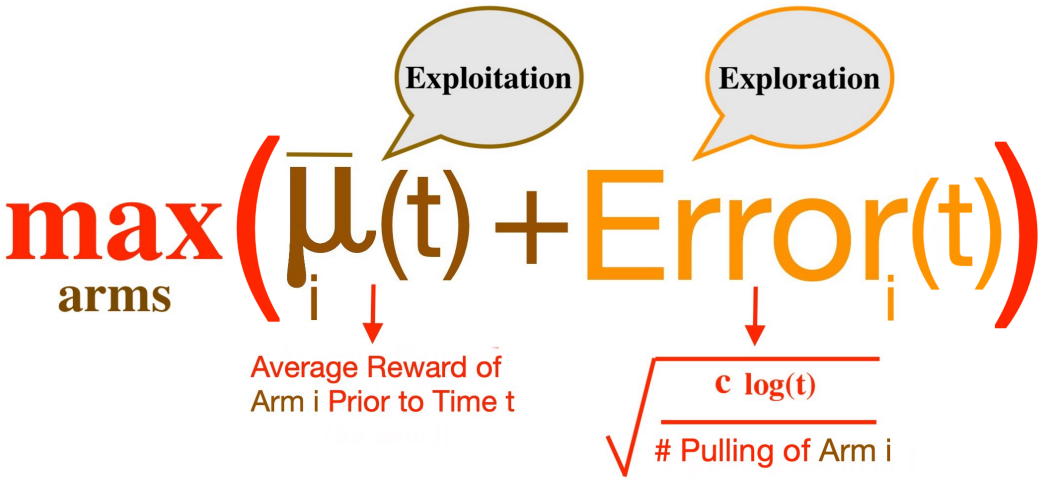
Upper Confidence Bound Bandit (UCB1)

1. Pull each of K arms once!
2. Pull the arm whose empirical average reward prior to time t plus its error has the maximum value and repeat this step!

Theorem (P. Auer, N. Cesa-Bianchi, and P. Fischer)

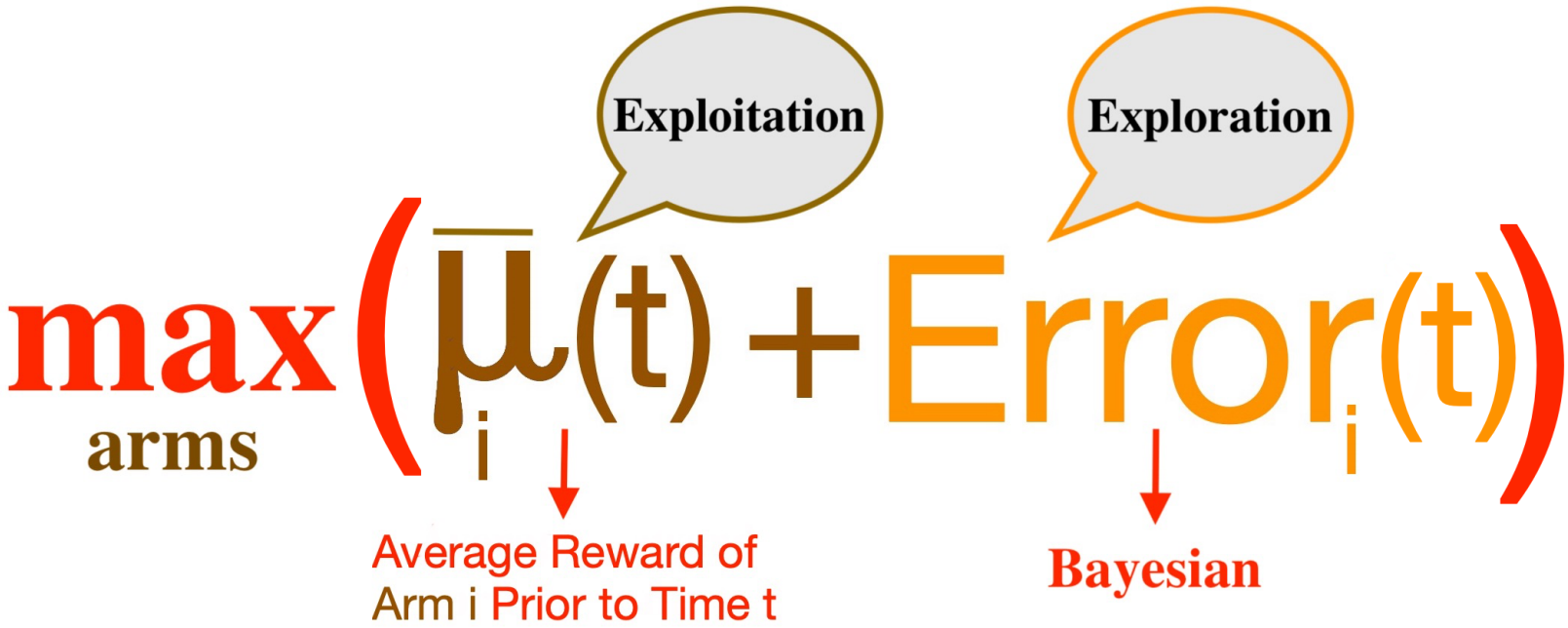
for all rounds $t \leq T$, Algorithm achieves regret:

$$R(T) = O(\sqrt{Kt \log T}).$$



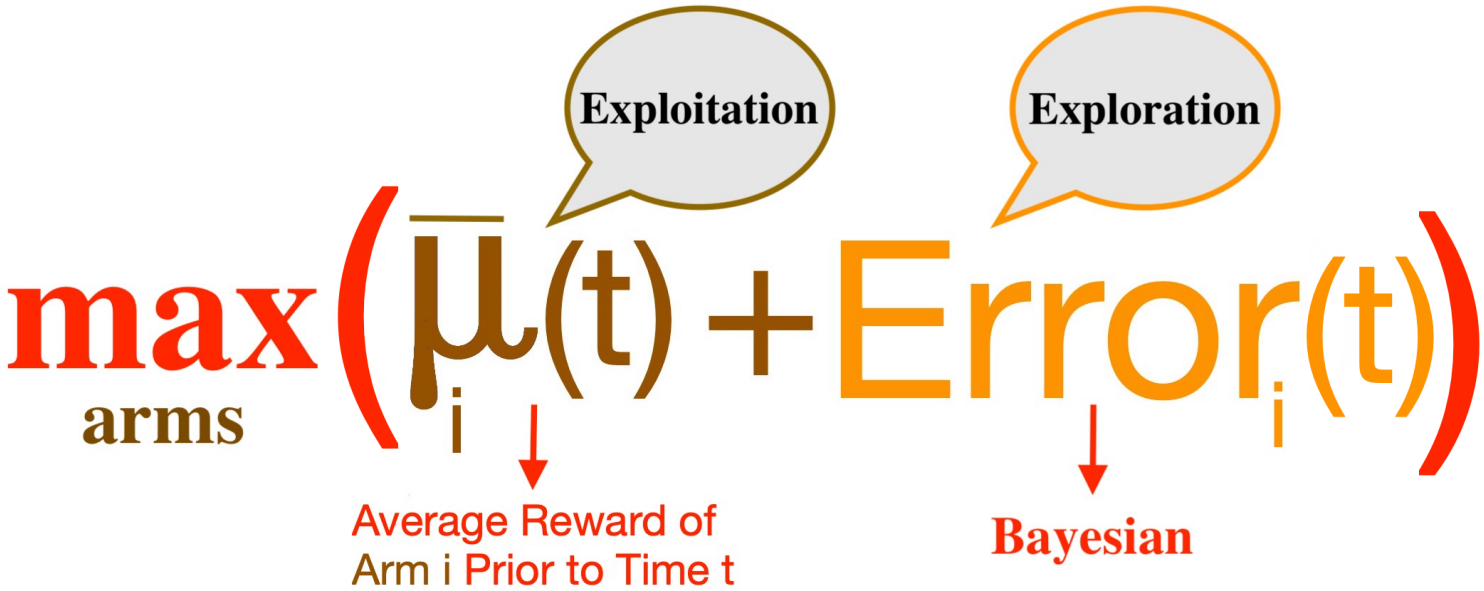
Bayesian (Based Error)

1. Pull each of K arms once!
2. Pull the arm whose empirical average reward prior to time t plus its error has the maximum value and repeat this step!



Bayesian Algorithms

1. Pull each of K arms once!
2. Pull the arm whose empirical average reward prior to time t plus its error has the maximum value and repeat this step!
3. Bayesian: $\max_i\{\bar{\mu}_i(t) + \bar{\sigma}_i(t) \times \Phi^{-1}(1-\alpha_t)\}$, where $\bar{\mu}_i(t)$ and $\bar{\sigma}_i(t)$ are the Bayesian mean, Bayesian variance. Also, $\Phi^{-1}(.)$ stands for quantile function of the standard normal random variable!



Finite Regret

Question: is it possible to introduce an algorithm with finite regret? NO!

Theorem (P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire)

Fix time horizon T and the number of arms K. For any bandit algorithm, there exists a problem instance such that

$$R(T) \geq \Omega(\sqrt{KT}).$$

Finite Regret

Question: is it possible to introduce an algorithm with finite regret? NO!

1. In practice, we are often happy to perform a task just good enough. For example, when driving to work we will be content with a strategy that will let us arrive just in time.
2. We only care about whether an arm with mean reward $\geq S$ is chosen, where S is the level of satisfaction we aim at.
3. Modify the classic notion of regret and consider, the satisficing (pseudo-)regret with respect to S (short S-regret) defined as

S-Regret:

$$R_S(T) = \sum_{t=1}^{T} \max\{S - \mu_{I_t}, 0\}$$

Regret:

$$R(T) = T\mu^* - \sum_{t=1}^{T} \mu_{I_t}$$

MONTAN UNIVERSITÄT LEOBEN

Satisfying Bandit

Theorem (T. Michel, H.H., R. Ortner)

If $S < \mu^\star$, there exists a constant C such that for any T, $R_S(T) \leq C$.

ALGORITHM

Input: K, S

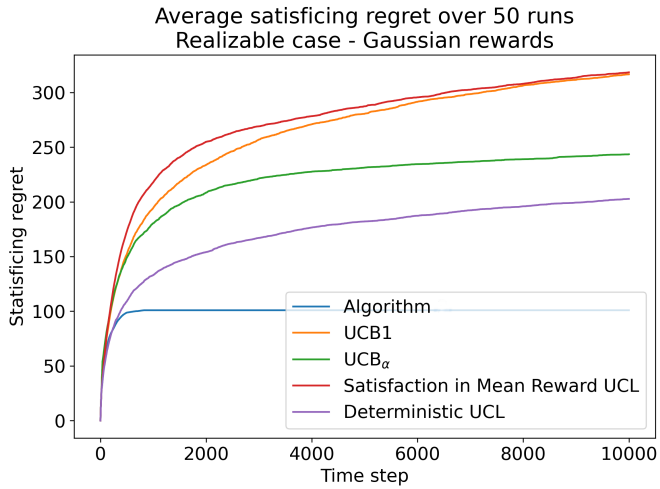
- Play each arm once.
- **For** each further step t **do**

  If $\exists$ arm i with $\hat{\mu}_i \geq S$ play

  $$A_t \in \operatorname*{argmax}_{1 \leq i \leq k} \left\{ \frac{\mathrm{UCB}_i(t) - \max\{S, \mathrm{LCB}_i(t)\}}{\beta_i(t)} \right\}.$$

  Else If $\exists$ arm i with $\mathrm{UCB}_i(t) \geq S$ choose A_t uniformly at random from $\{i | \, \mathrm{UCB}_i \geq S\}$.

  Else choose $A_t \in \operatorname*{argmax}_{1 \leq i \leq k} \mathrm{UCB}_i(t)$.



▸ Reference

Best Arm Identification: Two Arms

Exploration versus Exploitation

Based on the goal, the optimal algorithms may move towards more exploration than exploitation

- Assume that there are just two arms.
- The distribution of the rewards of each arm is Gaussian with unknown mean and the same variance.
- The goal is the best arm identification: minimize the error probability at time $2T$.

Algorithm:

1. Pull each arm T times!

MONTAN UNIVERSITÄT LEOBEN

Applications and Variants

Applications

1. K treatments for a given symptom (with unknown effect)
2. What treatment should be allocated to the next patient, based on responses observed on previous patients?

▸ Reference

Applications

1. K treatments for a given symptom (with unknown effect)
2. What treatment should be allocated to the next patient, based on responses observed on previous patients?

▸ Reference

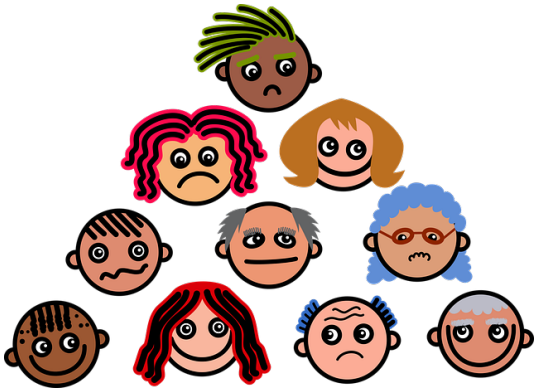
3. K adds that can be displayed
4. Which add should be displayed for a user, based on the previous clicks of previous (similar) users?

▸ Reference

MONTAN UNIVERSITÄT LEOBEN

Contextual Bandit

1. Assign to each person a feature or a context.
2. Imagine that each machine responds differently to each person.
3. You need to find the best strategy for the given context.
4. K treatments and K ads can be considered as contextual bandit.
5. Variants: Adversarial Bandit, Infinite-Armed Bandit, Non-Stationary Bandit, and so on

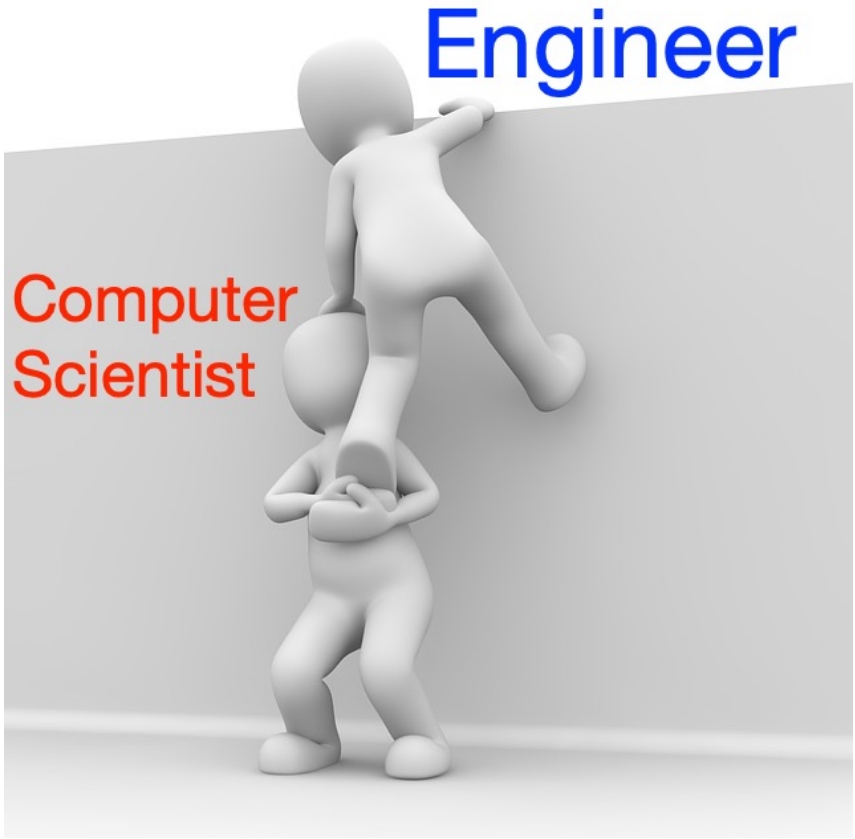


▸ Reference

▸ Reference

Ultimate Goal: For given problem, find algorithm that works well in practice (performance, efficiency), has favorable theoretical guarantees.

Engineer: Your theoretical results helped me so much about K adds!
Computer Scientist: You are welcome!



▸ Reference

Acknowledgement

Thanks for your attention!

I would like to show my greatest appreciation to Professor Ronald Ortner and Professor Xuding Zhu for the immeasurable amount of their support. Furthermore, they challenged me to push further and to learn more.

MONTAN UNIVERSITÄT LEOBEN