Chapter
Reinforcement Learning; Applications

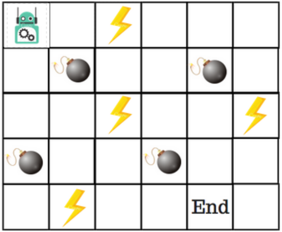
GE461: Introduction to Data Science

Cem Tekin
Bilkent University

Reinforcement learning (RL)

How should an agent interact with its environment in order to maximize its cumulative reward?

Example: Robot in a gridworld



S_0: initial state (position)

In each round t

- Take action A_t in state S_t (move 1 step left, right, up or down)
- Observe the next state S_{t+1}
- Collect reward R_{t+1} (-100 if bomb hit, 1 if power found, 100 if end reached, -1 otherwise)

Figure by Akshay Lambda from https://medium.com/free-code-camp/an-introduction-to-q-learning-reinforcement-learning-14ac0b4493cc

Reinforcement learning (RL)

Goal

Given discount rate $0 \leq \gamma \leq 1$ select actions to maximize

$$\text{(total return) } G_1 = R_1 + \gamma R_2 + \gamma^2 R_3 + \ldots = \sum_{k=0}^{\infty} \gamma^k R_{k+1}$$

Discount rate represents how much the agent cares about immediate rewards vs. future rewards

Policy π (method to select actions)

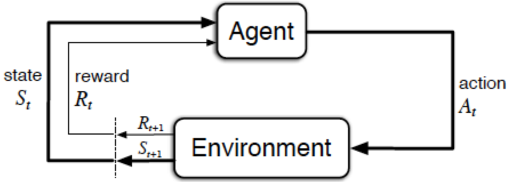
- History $\mathcal{H}_t = \{S_0, A_0, R_1, \ldots, S_{t-1}, A_{t-1}, R_t, S_t, A_t\}$ everything that happened by the end of round t
- Policy π maps past information to distributions over actions
- A_t sampled from $\pi(\cdot|\mathcal{H}_{t-1}, R_t, S_t)$
- π is deterministic if it puts all probability to a single action

What is a good model the environment?

Markov property

$$\Pr(R_{t+1} = r, S_{t+1} = s' | \mathcal{H}_t) = \Pr(R_{t+1} = r, S_{t+1} = s' | S_t, A_t)$$

General RL model



Some real-world applications

- Autonomous driving
- Personalized medicine
- Web advertising
- News, video, movie recommendation

Figure 3.1 from "Reinforcement Learning: An Introduction" by Sutton and Barto

Markov Decision Process (MDP)

A mathematical framework for modeling the interaction between agent and environment under Markov assumption

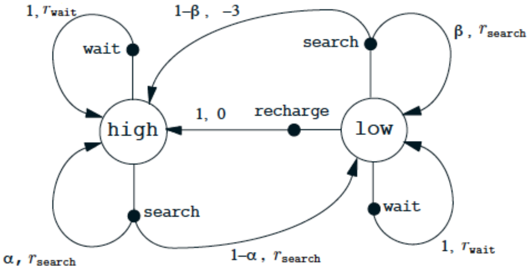
- Finite set of states: $\mathcal{S}$
- Finite set of actions: $\mathcal{A}$
- State transition probabilities:

$$p(s'|s,a) := \Pr(S_{t+1} = s' | S_t = s, A_t = a)$$

- Expected reward:

$$r(s,a,s') = \mathrm{E}[R_{t+1} | S_t = s, A_t = a, S_{t+1} = s']$$

Recycling robot example



- $\mathcal{S} = \{\text{high}, \text{low}\}$
- $\mathcal{A}(\text{high}) = \{\text{search}, \text{wait}\}$
- $\mathcal{A}(\text{low}) = \{\text{search}, \text{wait}, \text{recharge}\}$
- $r_{\text{search}} > r_{\text{wait}}$ [expected num. of cans collected by the robot]
- State transitions are random

Figure 3.3 from “Reinforcement Learning: An Introduction" by Sutton and Barto

Recycling robot example

s	s'	a	$p(s' \mid s, a)$	$r(s, a, s')$
high	high	search	α	$r_{\texttt{search}}$
high	low	search	$1-\alpha$	$r_{\texttt{search}}$
low	high	search	$1-\beta$	-3
low	low	search	β	$r_{\texttt{search}}$
high	high	wait	1	$r_{\texttt{wait}}$
high	low	wait	0	$r_{\texttt{wait}}$
low	high	wait	0	$r_{\texttt{wait}}$
low	low	wait	1	$r_{\texttt{wait}}$
low	high	recharge	1	0
low	low	recharge	0	0.

Table 3.1 from “Reinforcement Learning: An Introduction" by Sutton and Barto

Markov policies and the value function

General policy

A_t sampled from $\pi(\cdot|\mathcal{H}_{t-1}, R_t, S_t)$

Stationary Markov policy

A_t sampled from $\pi(\cdot|S_t)$

Total return after time t

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

State-value function for π

$$v_\pi(s) = \mathrm{E}_\pi[G_t|S_t = s] = \mathrm{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1}|S_t = s\right]$$

Action-value (Q) function for π

$$\begin{aligned} q_\pi(s,a) &= \mathrm{E}_\pi[G_t|S_t = s, A_t = a] \\ &= \mathrm{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1}|S_t = s, A_t = a\right] \end{aligned}$$

Optimal policy

π^* is optimal iff $v_{\pi^*}(s) \geq v_\pi(s)$ for all $s \in \mathcal{S}$ and π

Theorem [Puterman, 1994]
For infinite horizon discounted MDP there exists a deterministic stationary Markov policy that is optimal.

Optimal state-value function $v_*(s) = \max_\pi v_\pi(s)$

Optimal action-value (Q) function $q_*(s,a) = \max_\pi q_\pi(s,a)$

Bellman optimality equations

$$v_*(s) = \max_{a \in \mathcal{A}(s)} \underbrace{\mathrm{E}[R_{t+1} + \gamma v_*(S_{t+1}) | S_t = s, A_t = a]}_{q_*(s,a)}$$

$$q_*(s,a) = \mathrm{E}[R_{t+1} + \gamma \underbrace{\max_{a'} q_*(S_{t+1}, a')}_{v_*(S_{t+1})} | S_t = s, A_t = a]$$

Optimal policy

$$\pi^*(s) = \arg\max_a q_*(s,a) \text{ for all states } s$$

Computing the optimal policy (when state transition probabilities are known)

Value Iteration

(1) Start with an initial guess of the value functions $v_0(s)$, $s \in \mathcal{S}$ (e.g., set to zero)

(2) Compute the new value functions (at iteration $k+1$) by updating the value functions found at iteration k:

$$\begin{aligned} v_{k+1}(s) &= \max_a \mathrm{E}[R_{t+1} + \gamma v_k(S_{t+1}) | S_t = s, A_t = a] \\ &= \max_a \sum_{s'} p(s'|s,a)[r(s,a,s') + \gamma v_k(s')] \end{aligned}$$

(3) Repeat the above procedure until convergence, i.e., $||v_{k^*} - v_{k^*-1}|| \leq \epsilon$

(4) The final policy is

$$\pi(s) = \arg\max_a \left\{ \sum_{s'} p(s'|s,a) \left[r(s,a,s') + \gamma v_{k^*}(s') \right] \right\}$$

Computing the optimal policy (when state transition probabilities are known)

Value Iteration (with Q function)

(1) Start with an initial guess of the Q functions $q_0(s, a)$, $s \in \mathcal{S}$, $a \in \mathcal{A}$ (e.g., set to zero)

(2) Compute the new Q functions (at iteration $k + 1$) by updating the Q functions found at iteration k:

$$q_{k+1}(s, a) = \sum_{s'} p(s'|s, a) \left[r(s, a, s') + \gamma \max_{a'} q_k(s', a') \right]$$

(3) Repeat the above procedure until convergence

(4) We have $v_{k^*}(s) = \max_a q_{k^*}(s, a)$

(5) $\pi(s) = \arg\max_a q_{k^*}(s, a)$

Robot grid-world example for value iteration

`https://youtu.be/gThGerajccM`

- Goal location: high reward
- Freespace: small penalty
- Obstacles: very large penalty

Types of robots:

- Deterministic: Always moves in the direction of the dictated action
- Stochastic: Can also move in other directions with a positive probability

Learning the optimal policy (when state transition probabilities are unknown)

Estimate $q^*(s, a)$ in a data-driven manner. Recall that

$$q_*(s,a) = \mathrm{E}[R_{t+1} + \gamma \underbrace{\max_{a'} q_*(S_{t+1}, a')}_{v_*(S_{t+1})} | S_t = s, A_t = a]$$

Q learning

- Keep a table of Q value estimates: $Q(s, a)$ for $s \in \mathcal{S}$, $a \in \mathcal{A}$
- In round t: $S_t \underbrace{\rightarrow}_{\text{How?}} A_t \rightarrow (S_{t+1}, R_{t+1})$
- Form sample estimate: $\hat{Q}(S_t, A_t) = R_{t+1} + \gamma \max_{a'} Q(S_{t+1}, a')$
- Update Q-value of (S_t, A_t)

$$Q(S_t, A_t) \leftarrow (1-\alpha) Q(S_t, A_t) + \underbrace{\alpha}_{\text{learning rate}} \hat{Q}(S_t, A_t)$$

Convergence If all (s, a) pairs are selected infinitely many times

$$Q(s, a) \rightarrow q_*(s, a) \text{ with probability } 1$$

How to choose A_t given S_t?

Option 1: Greedy

$$A_t = \arg\max_a Q(S_t, a)$$

Always exploits. No exploration. Might stuck in suboptimal

Option 2: ϵ-greedy

- Toss a coin C_t with $\Pr(C_t = H) = \epsilon$
- If $C_t = H$, then sample A_t uniformly randomly from action set (explore)
- If $C_t = T$, then $A_t = \arg\max_a Q(S_t, a)$ (exploit)

Option 3: Boltzmann exploration

$$A_t \sim \Pr(\cdot|S_t) \text{ such that } \Pr(A_t = a|S_t) = \frac{e^{Q(S_t,a)}}{\sum_{a'} e^{Q(S_t,a')}}$$

Explores implicitly

Deep Q Network Learning to Play Atari Game

https://youtu.be/cjpEIotvwFY

The multi-armed bandit problem

Gambling in a casino with K arms (slot machines)

In each round t

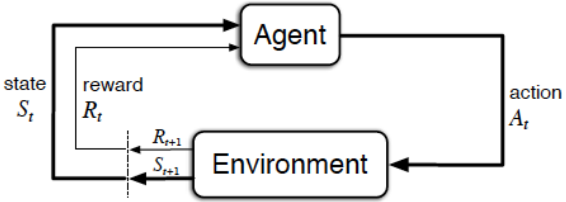
- Play an arm A_t
- Collect its random reward $R_{A_t,t}$ that comes from an unknown distribution

Goal: Maximize expected total reward $\mathbb{E}\left[\sum_t R_{A_t,t}\right]$

Multi-armed bandits and reinforcement learning

General RL framework

- Repeated interaction over time $t = 1, 2, \ldots$
- S_t: state at time t. A_t: action at time t. R_t: reward at time t

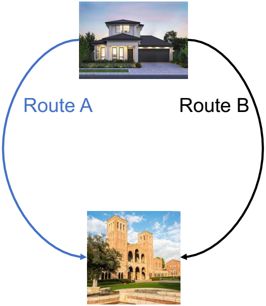


- General RL: S_{t+1} depends on past actions and states (e.g., Markov model)
- K-armed stochastic bandit: one state
- More structure $\Rightarrow$ more specialized algorithms & faster learning/convergence & rigorous optimality guarantees

Sequential decision-making under uncertainty: navigation

How to go from home to school?

- Day 1: Route A. Travel time: 20 min
- Day 2: Route A. Travel time: 40 min
- Day 3: Route B. Travel time: 25 min
- Day 4: ?



Travel times are uncertain

Want to

- Minimize $\sum$ travel times

Sequential decision-making under uncertainty: recommender system

Pool of items $\{A, B, C, \ldots\}$

Users arrive sequentially over time ($t = 1, 2, \ldots$)

What should we recommend to maximize number of clicks

- User 1: Item A. Clicked
- User 2: Item A. Not clicked
- User 3: Item B. Clicked
- User 4: ?

User behavior is uncertain

Sequential decision-making under uncertainty: cognitive communications

Channels with time varying qualities $\{A, B, C, \ldots\}$

Time-slotted communication ($t = 1, 2, \ldots$)

Which channels should be selected to maximize throughput

- Time slot 1: Channel A. Successful transmission
- Time slot 2: Channel A. Failed transmission
- Time slot 3: Channel B. Successful transmission
- Time slot 4: ?

Channel gains are unknown, their distributions are unknown

How to play the game

1. Know the the environment class $\mathcal{E}$

- Arm set $\mathcal{A} = \{1, \ldots, K\}$
- Reward from arm a is sampled from unknown F_a, independent of other arms

This is called *stochastic K-armed bandit*

- Assume: $R_{a,t} \in [0, 1]$ bounded support (alternatives: Bernoulli, Gaussian, subGaussian, heavy tailed)

2. Construct a policy

- History $\mathcal{H}_t = \{A_1, R_{A_1,1}, \ldots, A_{t-1}, R_{A_{t-1},t-1}\}$
- Policy π : histories $\rightarrow$ distributions over $\mathcal{A}$

3. Play according to your policy

- Play $A_t \sim \pi(\cdot|\mathcal{H}_t)$
- Observe $R_{A_t,t} \sim F_{A_t}$
- Update $\mathcal{H}_{t+1} = \mathcal{H}_t \cup \{A_t, R_{A_t,t}\}$

Regret of a policy

Expected reward of arm a : $\mu_a = \mathbb{E}[R_{a,t}]$

- Always select the best arm $a^* = \arg\max_a \mu_a$
- Highest expected reward: $\mu^* = \mu_{a^*}$
- Highest cumulative expected reward in T rounds: $T \times \mu^*$

Regret

$$\text{Reg}_\pi(T) = T \times \mu^* - \sum_{t=1}^{T} \mu_{A_t}$$

Fact

$$\text{Max}_\pi \mathbb{E}\left[\sum_{t=1}^{T} R_{A_t,t}\right] = \text{Min}_\pi \mathbb{E}\left[\text{Reg}_\pi(T)\right]$$

What is a good policy?

For all bandit instances in $\mathcal{E}$ (e.g., all K-armed bandits with independent arm rewards in $[0, 1]$)

$$\lim_{T \to \infty} \frac{\mathbb{E}\left[\mathrm{Reg}_\pi(T)\right]}{T} = 0$$

Examples: $\mathbb{E}\left[\mathrm{Reg}_\pi(T)\right] = O(\sqrt{T})$, $\mathbb{E}\left[\mathrm{Reg}_\pi(T)\right] = O(\log T)$

Since expected rewards are unknown, a good policy should

- Explore arms to discover the best
- Exploit the arm that is believed to be the best
- Be computationally efficient

Regret lower bound

Consistent policy

π is consistent if for all $\{F_a\}_{a=1}^{K} \in \mathcal{E}$ and $p > 0$

$$\lim_{T\to\infty} \frac{\mathbb{E}[\mathrm{Reg}_\pi(T)]}{T^p} = 0$$

Asymptotic lower bound*

Let $\mathcal{E}$ be class of bandits with single parameter exponential family of reward distributions (e.g., $F_a = \mathrm{Ber}(\theta_a)$, $R_{a,t} \in \{0, 1\}$). For a consistent policy π regret grows at least logarithmically over time.

$$\liminf_{T\to\infty} \frac{\mathbb{E}[\mathrm{Reg}_\pi(T)]}{\log T} \geq \sum_{a:\mu_a<\mu^*} \frac{\mu^* - \mu_a}{\mathrm{KL}(a, a^*)}$$

Minimum achievable regret $O(\log T)$

*Lai and Robbins 1985: Asymptotically efficient adaptive allocation rules.

Greedy policy

Sample mean reward collected from arm a by the end of round $t-1$: $\hat{\mu}_{a,t-1}$

Initially
Sample each arm once

At each round $t > K$
Select $A_t = \arg\max_a \hat{\mu}_{a,t-1}$

Example with $K = 2$ arms
Bernoulli rewards, $\mu_i = 0.9$, $\mu_j = 0.8$

t	$\hat{\mu}_{i,t-1}$	$\hat{\mu}_{j,t-1}$	A_t	$r_{A_t,t}$
1			i	1
2			j	1
3	1	1	i	0
4	$1/2$	1	j	1
5	$1/2$	1	j	1
6	$1/2$	1	?	?

Might get stuck in arm j which is suboptimal

ϵ_t-greedy policy

A sequence of exploration probabilities $\{\epsilon_t\}$

Empirical best arm $\hat{a}_t^* = \arg\max_a \hat{\mu}_{a,t-1}$

Initially

- Sample each arm once

At each round $t > K$

- Explore with probability ϵ_t

$$\text{Select } A_t \text{ randomly from } \{1, 2, \ldots, K\}$$

- Exploit with probability $1 - \epsilon_t$

$$\text{Select } A_t = \hat{a}_t^*$$

Regret of ϵ_t-greedy algorithm

Let $\Delta_a = \mu^* - \mu_a$ suboptimality gap
Let $\Delta_{\min} = \min_{a:\mu_a<\mu^*} \Delta_a$
Tune exploration probabilities

$$\epsilon_t = \frac{cK}{\Delta^2_{\min} t}, c > 0$$

Regret bound*

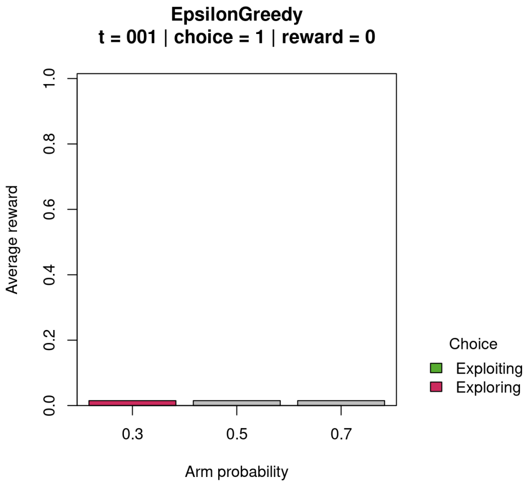
$$\mathbb{E}\left[\text{Reg}_{\epsilon_t-\text{greedy}}(T)\right] \leq c' \times \sum_{a=1}^{K} \left(\Delta_a + \frac{\Delta_a}{\Delta^2_{\min}} \log \max\left\{e, \frac{T\Delta^2_{\min}}{K}\right\}\right)$$
$$= O\left(\frac{K \log T}{\Delta^2_{\min}}\right)$$

Takeaways

- Exploration achieved by randomization
- Need careful tuning
- Uniform exploration

*Auer et al. 2002: Finite-time analysis of the multiarmed bandit problem.

ϵ_t-greedy in action



Upper Confidence Bound (UCB) policy

Initially

- Sample each arm once

At each round $t > K$

1. Calculate optimistic estimate of arm a

$$\underbrace{g_{a,t}}_{\text{index}} = \underbrace{\hat{\mu}_{a,t-1}}_{\text{sample mean}} + \underbrace{\sqrt{\frac{2 \log t}{N_{a,t-1}}}}_{\text{exploration bonus}}$$

2. Select the optimistic best arm

$$a_t = \arg\max_a \; g_{a,t}$$

Fact: $g_{a,t}$ is an upper confidence bound for μ_a, i.e., with high probability $g_{a,t} \geq \mu_a$ for all arms

Auer et al. 2002: Finite-time analysis of the multiarmed bandit problem.

Regret of UCB policy

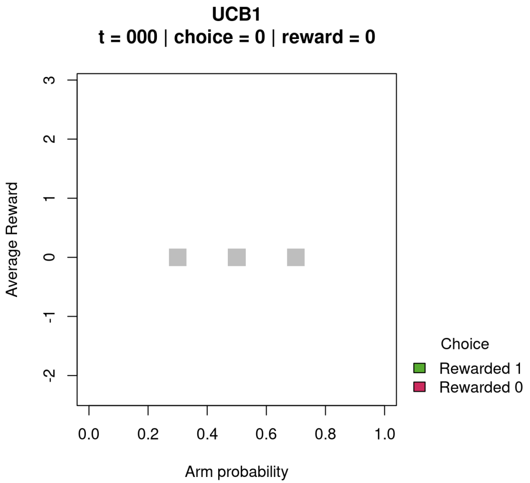
Regret bound

$$\mathbb{E}\left[\mathrm{Reg}_{\mathrm{UCB}}(T)\right] \leq 8 \sum_{a:\mu_a<\mu_*} \frac{\log T}{\mu_* - \mu_a} + \left(1 + \frac{\pi^2}{3}\right) \sum_a (\mu_* - \mu_a)$$
$$= O\left(\sum_{a:\mu_a<\mu^*} \frac{\log T}{\Delta_a} \right)$$

Takeaways

- Exploration achieved by *optimism under uncertainty*
- Adaptive exploration
- Deterministic policy

UCB in action



Regret analysis for UCB

Regret decomposition

Recall: $\Delta_a = \mu^* - \mu_a$ suboptimality gap

$N_{a,t} = \sum_{s=1}^{t} \mathbb{I}(A_s = a)$ number of plays of arm a by round t

$$\mathbb{E}\left[\text{Reg}_\pi(T)\right] = T\mu^* - \mathbb{E}\left[\sum_{t=1}^{T} \mu_{A_t}\right] \quad (1)$$

$$= \sum_{t=1}^{T} \mu^* - \mathbb{E}\left[\sum_{t=1}^{T}\sum_{a=1}^{K} \mu_a \mathbb{I}(A_t = a)\right] \quad (2)$$

$$= \mathbb{E}\left[\sum_{a=1}^{K}(\mu^* - \mu_a)\sum_{t=1}^{T} \mathbb{I}(A_t = a)\right] \quad (3)$$

$$= \sum_{a=1}^{K} \Delta_a \mathbb{E}[N_{a,t}] \quad (4)$$

Regret analysis for UCB

Recall regret decomposition

$$\mathbb{E}\left[\mathrm{Reg}_{\pi}(T)\right] = \sum_{a=1}^{K} \Delta_a \mathbb{E}[N_{a,t}]$$

Bounding $\mathbb{E}[N_{a,t}]$ for suboptimal arms

$$N_{a,t} = 1 + \sum_{t=K+1}^{T} \mathbb{I}(A_t = a) \tag{5}$$

$$= 1 + \sum_{t=K+1}^{T} \mathbb{I}(A_t = a, N_{a,t-1} \geq m) + \sum_{t=K+1}^{T} \mathbb{I}(A_t = a, N_{a,t-1} < m) \tag{6}$$

$$\leq m + \sum_{t=K+1}^{T} \mathbb{I}(A_t = a, N_{a,t-1} \geq m) \tag{7}$$

$$\leq m + \sum_{t=K+1}^{T} \mathbb{I}(g_{a,t} \geq g_{a^*,t}, N_{a,t-1} \geq m) \tag{8}$$

Regret analysis for UCB

When $N_{a,t-1} \geq m = \lceil \frac{8 \log T}{(\mu^* - \mu_a)^2} \rceil$, $g_{a,t} \geq g_{a^*,t}$ happens when

$$\text{Either } \underbrace{\hat{\mu}_{a,t-1} - \sqrt{\frac{2 \log t}{N_{a,t-1}}} \geq \mu_a}_{\text{LCB}_t \text{ fails}} \text{ or } \underbrace{\hat{\mu}_{a^*,t-1} + \sqrt{\frac{2 \log t}{N_{a^*,t-1}}} \leq \mu^*}_{\text{UCB}_t \text{ fails}}$$

Assuming that $N_{a,t-1}$ and $N_{a^*,t-1}$ are fixed (not random), Hoeffding's inequality implies that

$$\Pr(\text{LCB}_t \text{ fails}) \leq t^{-4}, \quad \Pr(\text{UCB}_t \text{ fails}) \leq t^{-4}$$

Actual proof requires taking a union bound over possible realizations of $N_{a,t-1}$ and $N_{a^*,t-1}$.

Finally,

$$\begin{aligned}
\mathbb{E}\left[N_{a,T}\right] &\leq m + \mathbb{E}\left[\sum_{t=K+1}^{T} \mathbb{I}(g_{a,t} \geq g_{a^*,t}, N_{a,t-1} \geq m)\right] \\
&= m + \sum_{t=K+1}^{T} \Pr(g_{a,t} \geq g_{a^*,t}, N_{a,t-1} \geq m) \\
&\leq m + \frac{\pi^2}{3} = \lceil \frac{8 \log T}{(\mu^* - \mu_a)^2} \rceil + \frac{\pi^2}{3}
\end{aligned}$$

Thompson (posterior) sampling

Bayesian algorithm (William R. Thompson in 1933)

1 Start with prior over bandit instances $p(\{F_a\}_{a=1}^{K})$

2 Compute posterior distribution of the optimal arm $p(a^*|\mathcal{H}_t)$

3 $A_t \sim p(a^*|\mathcal{H}_t)$

Thompson (posterior) sampling

Bayesian algorithm (William R. Thompson in 1933)

1. Start with prior over bandit instances $p(\{F_a\}_{a=1}^K)$
2. Compute posterior distribution of the optimal arm $p(a^*|\mathcal{H}_t)$
3. $A_t \sim p(a^*|\mathcal{H}_t)$

Equivalently

1. Start with prior over bandit instances $p(\{F_a\}_{a=1}^K)$
2. Compute posterior over bandit instances $p(\{F_a\}_{a=1}^K|\mathcal{H}_t)$
3. Sample a bandit instance $\{\hat{F}_a\}_{a=1}^K \sim p(\{F_a\}_{a=1}^K|\mathcal{H}_t)$
4. $A_t = \arg\max_a \mu(\hat{F}_a)$

Thompson sampling for Bernoulli bandits

Bernoulli bandits $F_a = \text{Ber}(\theta_a)$, $R_{a,t} \in \{0,1\}$

Prior distribution $p(\{F_a\}_{a=1}^K) = \prod_{a=1}^K p(F_a)$, $p(F_a) = \text{Beta}(1,1)$

Posterior distribution $p(F_a|\mathcal{H}_t) = \text{Beta}(1+\alpha_{a,t-1}, 1+\beta_{a,t-1})$

- $\alpha_{a,t-1}$: number successes (1) from arm a by end of $t-1$
- $\beta_{a,t-1}$: number failures (0) from arm a by end of $t-1$

At each round t

1. Sample $\tilde{\mu}_{a,t}$ from $\text{Beta}(1+\alpha_{a,t-1}, 1+\beta_{a,t-1})$ (posterior)
2. Select $A_t = \arg\max_a \tilde{\mu}_{a,t}$
3. Observe $R_{A_t,t} \in \{0,1\}$
4. $\alpha_{A_t,t} = \alpha_{A_t,t-1} + R_{A_t,t}$, $\beta_{A_t,t} = \beta_{A_t,t-1} + 1 - R_{A_t,t}$

Regret bound for Thompson sampling

For Bernoulli bandits*, for every $\epsilon > 0$

$$\mathbb{E}\left[\mathrm{Reg}_{\mathrm{TS}}(T)\right] \leq (1+\epsilon) \sum_{a:\mu_a<\mu^*} \frac{(\log T + \log\log T)}{\mathrm{KL}(a, a^*)} \Delta_a + const$$

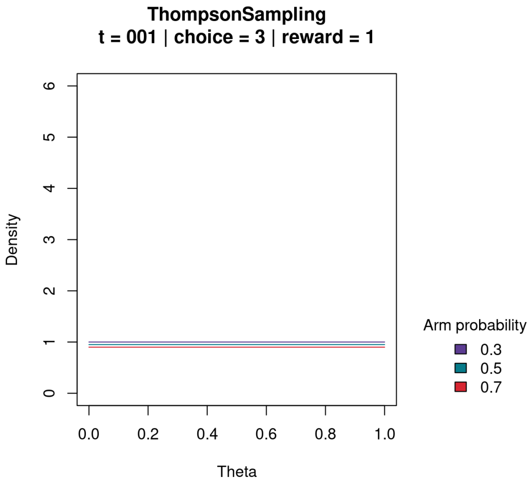
$$= O\Big(\sum_{a:\mu_a<\mu^*} \frac{\log T}{\Delta_a}\Big)$$

Takeaways

- Exploration achieved by sampling from posterior
- Adaptive exploration
- Randomized policy

*Kaufmann et al. 2012 "Thompson sampling: An asymptotically optimal finite-time analysis"

Thompson sampling in action



Empirical comparison

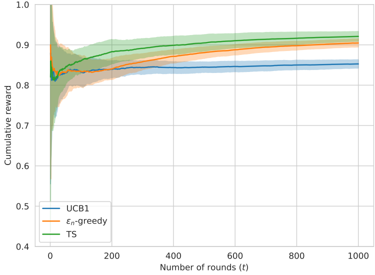


Figure: Average reward (tuned ϵ_t-greedy)

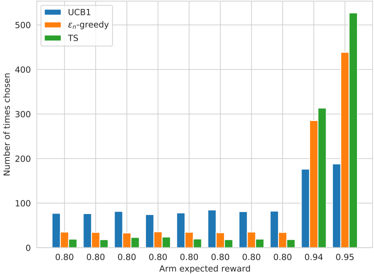


Figure: Average number of times each arm was played by the end of the simulation.

Summary

1 Studied stochastic K-armed bandit.
   - $R_{a,t} \sim F_a$ (unknown), indep. of other arms

2 Any consistent policy incurs at least $O(\log T)$ regret

3 Following policies that can achieve $O(\log T)$ regret

ϵ_t-greedy

- Explores with probability ϵ_t
- Uniformly explores all arms
- $O(\frac{K \log T}{\Delta_{\min}^2})$ regret (with tuned ϵ_t)

UCB

- Explores by being optimistic
- Adaptively explores
- $O(\sum_{a:\mu_a<\mu^*} \frac{\log T}{\Delta_a})$ regret

Thompson sampling

- Explores by sampling from posterior
- Adaptively explores
- $O(\sum_{a:\mu_a<\mu^*} \frac{\log T}{\Delta_a})$ regret