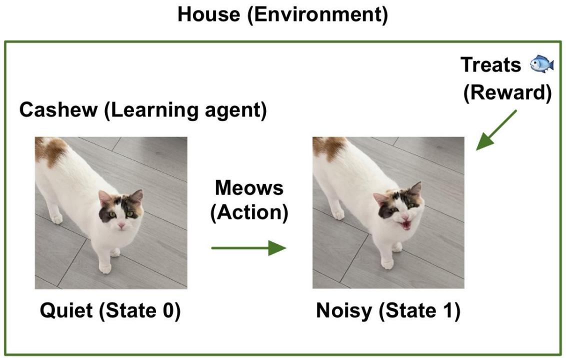
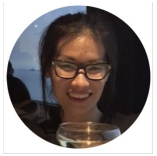


Optimistic Thompson Sampling

Strategic Exploration in Bandits and Reinforcement Learning

Master's Thesis Presentation
Helen Zhang



Bingshan Hu
University of Alberta, University of British Columbia, Amii

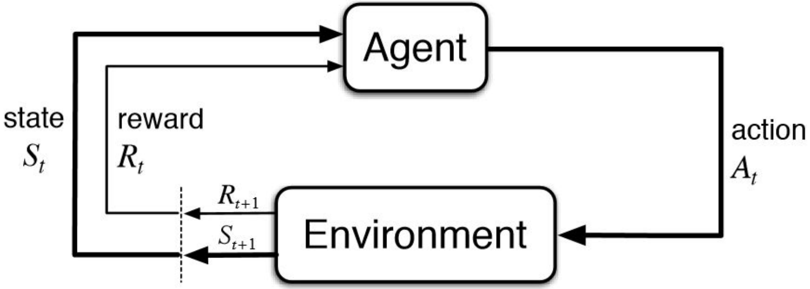
Tianyue H. Zhang
University of British Columbia

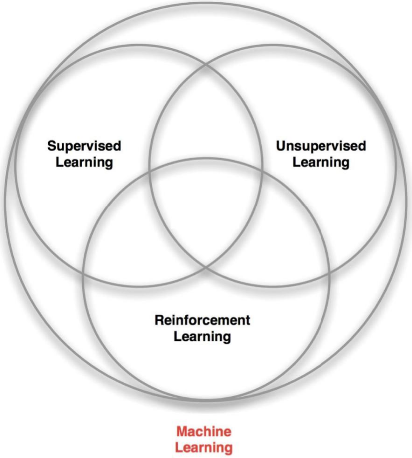
Nidhi Hegde
CIFAR Chair, University of Alberta, Amii Fellow

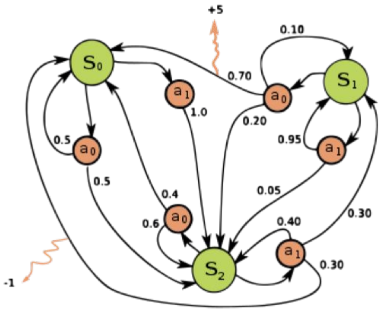
Mark Schmidt
CIFAR Chair, University of British Columbia, Amii Fellow

39th Conference on Uncertainty in Artificial Intelligence
Pittsburgh, PA, USA
31st July – 4th August, 2023
uai2023

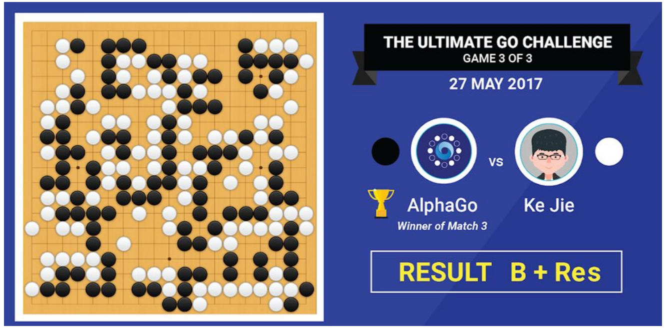
Reinforcement Learning (RL)





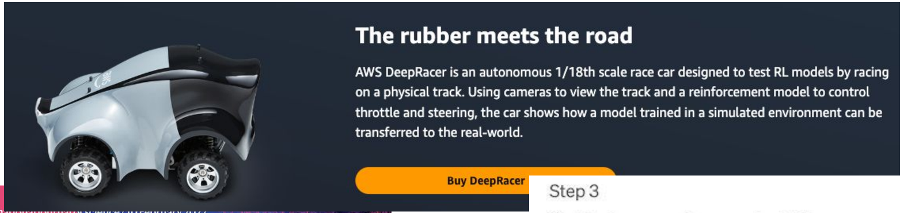


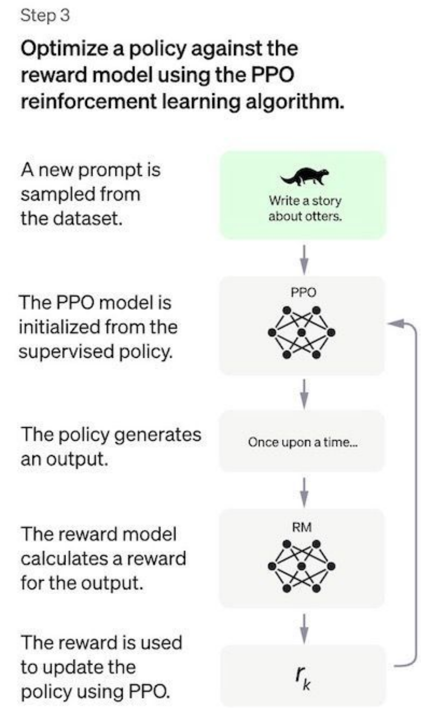
1950s

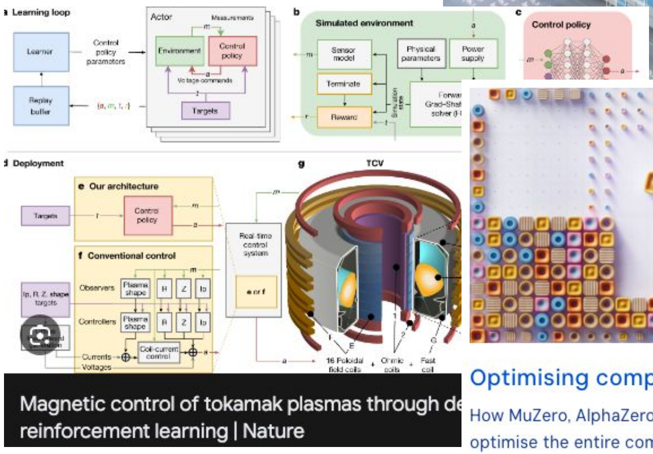


2017

Recent RL Applications







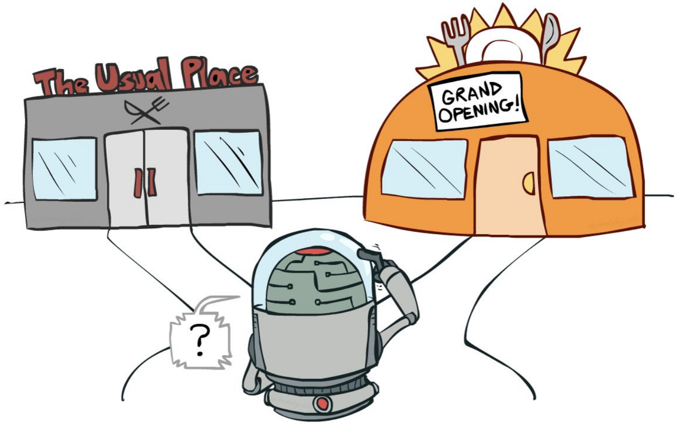
Optimising computer systems

How MuZero, AlphaZero, and AlphaDev are helping optimise the entire computing ecosystem that powers our world of devices.

Exploration-Exploitation Tradeoff

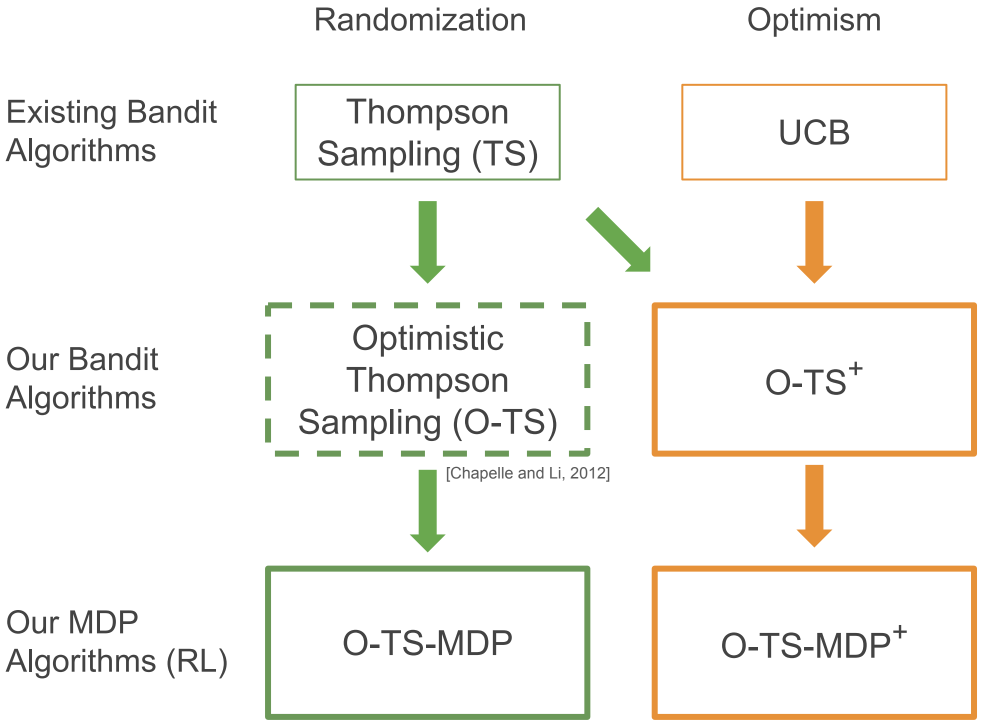
Exploitation
Take actions with high empirical reward to gain pay-off

Exploration
Take less observed actions to gather information



💰 Data is limited/expensive
🧐 Environment dynamics and reward is complex/unknown

Contributions



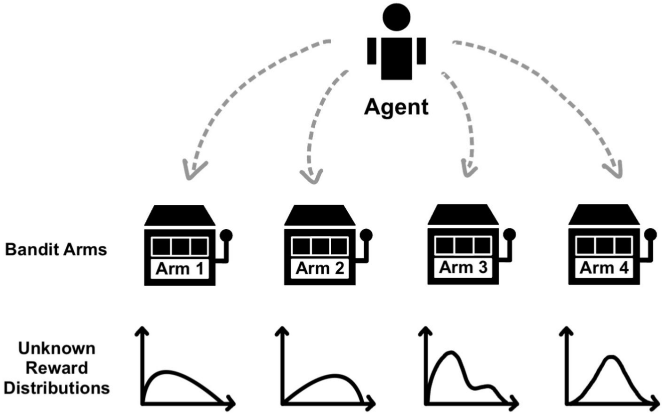
Bandits

Stochastic Multi-armed Bandits

Bandit Environment:

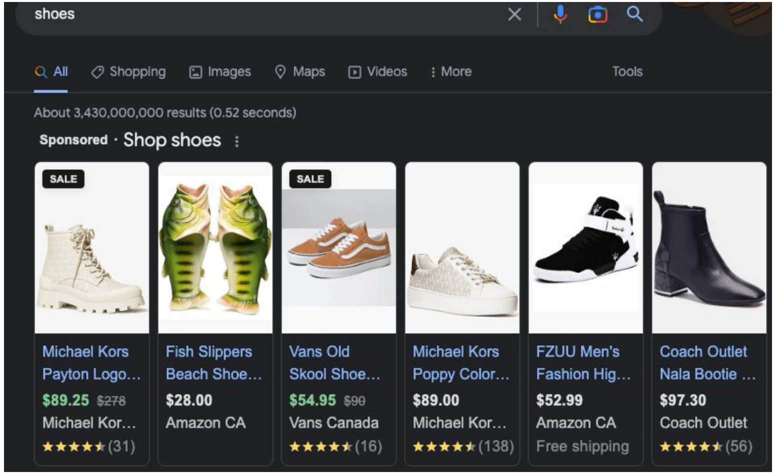
- A finite time horizon: T
- Number of arms : K
- Each arm has an unknown reward distribution with mean $\mu_1, \mu_2, \ldots, \mu_k$

Decision-making under uncertainty



For each time step t = 1,2,...,T:

- Agent pulls an arm $J_t = j, j \in [K]$
- Agent observes a random reward $X_j(t)$ with mean μ_j



Reward vs Regret

- Objective: Maximize cumulative reward
- But this doesn't tell you whether a policy is optimal

Regret: the difference between the reward from the played arm at each round, and the best possible reward

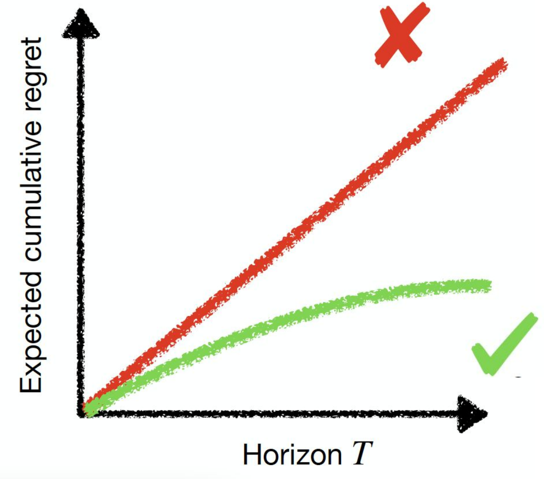
Reward of best arm

Reward of played arm

$$\mathcal{R}(T;\Theta) = \mathbb{E}\left[\sum_{t=1}^{T}\left(\max_{j\in[K]}\mu_j - \mu_{J_t}\right)\right] = \sum_{t=1}^{T}\mathbb{E}\left[\Delta_{J_t}\right]$$

$$\Delta_j = \mu_* - \mu_j$$

Sub-optimality gap



Goal: minimize (pseudo-) expected cumulative regret
(equivalent to maximize expected cumulative reward)

Uncertainty-Driven Exploration

A commonly used strategy: ε-greedy (Not optimal!)

We want to explore the arms that we are <u>uncertain (less observed)</u>

For an arm j, up to round t-1, we have:

- Number of observation: $O_j(t-1)$
- Empirical estimation of the reward: $\widehat{\mu}_{j,O_j(t-1)} = \frac{\sum_{\tau=1}^{t-1} \mathbb{1}[J_\tau = j] X_j(\tau)}{O_j(t-1)}$

Two exploration algorithms:

- Upper confidence bound (UCB)
- Thompson sampling

UCB vs TS

UCB: *Optimism in the face of uncertainty*

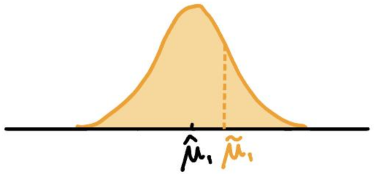
- Compute the empirical mean of each arm and a confidence interval;
- Use the upper confidence bound as a proxy for goodness of arm.

Algorithm 3.3 UCB

For each round $t = 1, 2, \dots, T$:
for each arm j **do**
  Set $\bar{\mu}_j \leftarrow \hat{\mu}_{j,O_j(t-1)} + \sqrt{\frac{2\ln(t)}{O_j(t-1)}}$
end for
Pull arm $J_t \leftarrow \arg\max_{j\in\mathcal{A}} \bar{\mu}_j(t)$

Bonus term

TS: *"Randomly take action according to the probability you believe it is the optimal action" - Thompson 1933*



- Compute the empirical mean of each arm and build a posterior distribution;
- Draw a random sample as a proxy for goodness of arm.

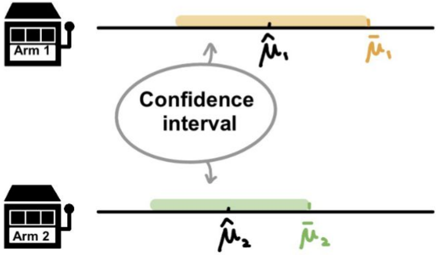
Algorithm 3.4 TS with Gaussian priors

For each round $t = 1, 2, \dots, T$:
for each arm j **do**
  Draw $\tilde{\mu}_j(t) \sim \mathcal{N}\left(\hat{\mu}_{j,O_j(t-1)}, \frac{1}{O_j(t-1)}\right)$
end for
Pull arm $J_t \leftarrow \arg\max_{j\in\mathcal{A}} \tilde{\mu}_j(t)$

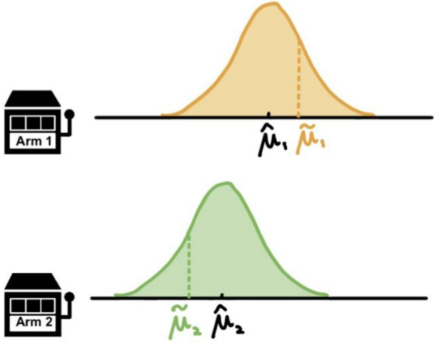
Variance

UCB vs TS

Exploitation

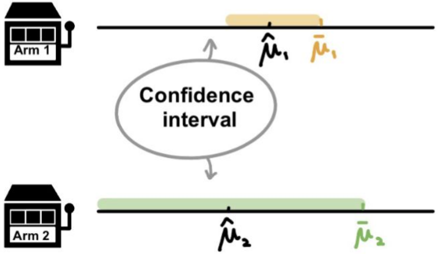


UCB

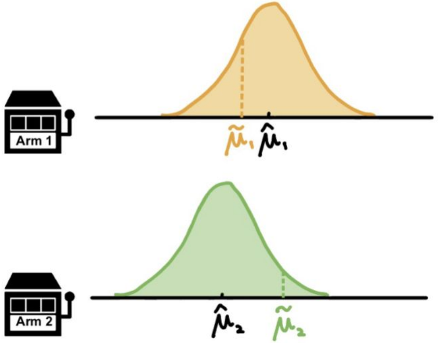


Thompson sampling

Exploration

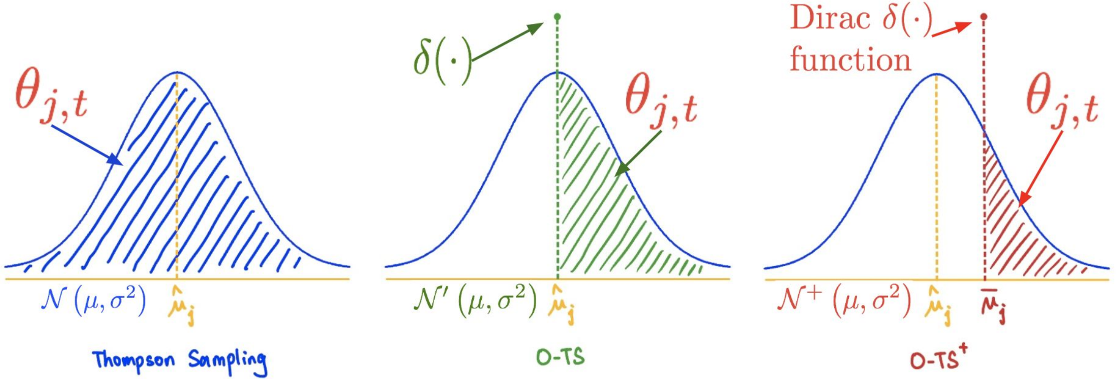


UCB



Thompson sampling

O-TS and O-TS$^+$



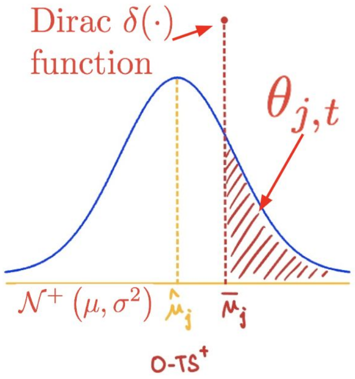
Algorithm 3.5 O-TS

1: For each round $t = 1, 2, ..., T$:
2: **for** each arm j **do**
3: Draw $\tilde{\mu}_j(t) \sim \mathcal{N}\left(\hat{\mu}_{j,O_j(t-1)}, \frac{1}{O_j(t-1)}\right)$
Set $\tilde{\mu}'_j(t) \leftarrow \max\{\tilde{\mu}_j(t), \hat{\mu}_j(t)\}$
4: **end for**
5: Pull arm $J_t \leftarrow \arg\max_{j \in \mathcal{A}} \tilde{\mu}'_j(t)$

Algorithm 3.6 O-TS$^+$

1: For each round $t = 1, 2, ..., T$:
2: **for** each arm j **do**
3: Draw $\tilde{\mu}_j(t) \sim \mathcal{N}\left(\hat{\mu}_{j,O_j(t-1)}, \frac{1}{O_j(t-1)}\right)$
Set $\bar{\mu}_j \leftarrow \hat{\mu}_{j,O_j(t-1)} + \sqrt{\frac{3\ln(t)}{O_j(t-1)}}$
Set $\tilde{\mu}'_j(t) \leftarrow \max\{\tilde{\mu}_j(t), \bar{\mu}_j(t)\}$
4: **end for**
5: Pull arm $J_t \leftarrow \arg\max_{j \in \mathcal{A}} \tilde{\mu}'_j(t)$

Proof Sketch



Hoeffding's inequality, w.h.p. :

$$\bar{\mu}_{j,O_j(t-1)} = \hat{\mu}_{j,O_j(t-1)} + \sqrt{\frac{3\ln(t)}{O_j(t-1)}} \geq \mu_j, \forall j \in [K]$$

$$\begin{aligned}
\Delta_{J_t} &= \mu_* - \mu_{J_t} \\
\text{UCB} \rightarrow \quad &\leq \bar{\mu}_{*,O_*(t-1)} - \mu_{J_t} \\
\text{Clipped Gaussian} \rightarrow \quad &\leq \tilde{\mu}'_{*,t} - \mu_{J_t} \\
\text{Arm J is pulled} \rightarrow \quad &\leq \underbrace{\tilde{\mu}'_{J_t,t} - \bar{\mu}_{J_t,O_{J_t}(t-1)}}_{\text{deviation bound of data-dependent distribution}} + \underbrace{\bar{\mu}_{J_t,O_{J_t}(t-1)} - \mu_{J_t}}_{\text{UCB}}
\end{aligned}$$

To upper bound $\mathbb{E}[\Delta_{J_t}]$, we have $\mathbb{E}\left[\tilde{\mu}'_{J_t,t} - \bar{\mu}_{J_t,O_{J_t}(t-1)}\right] \leq O\left(\sqrt{\frac{\ln(T)}{O_{J_t}(t-1)}}\right)$

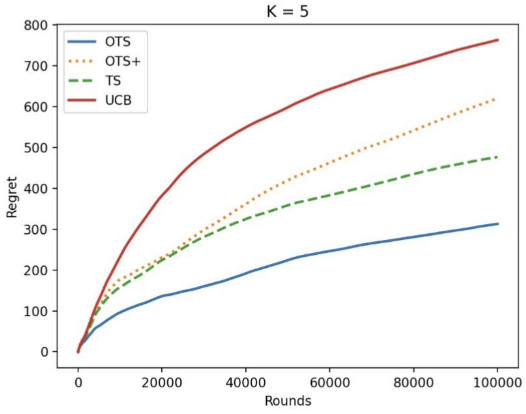
Results

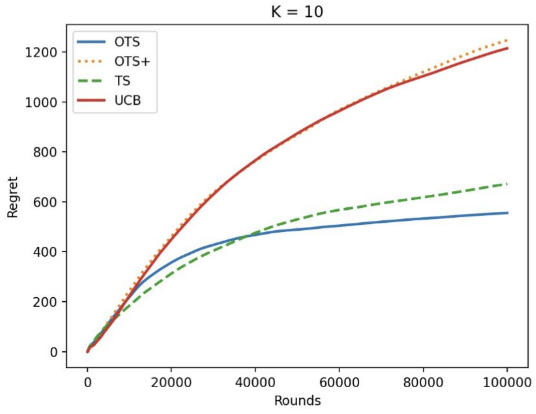
Regret:

UCB1 [Auer et al., 2002a]	$\mathbf{O}(\sqrt{KT\ln T})$
TS (Gaussian) [Agrawal and Goyal, 2017]	$\mathbf{O}(\sqrt{KT\ln K})$
O-TS	$\mathbf{O}(\sqrt{KT\ln K})$
O-TS$^+$	$\mathbf{O}(\sqrt{KT\ln T})$

All algorithms achieves (order-)optimal problem dependent regret $\mathbf{O}(\frac{K\ln T}{\Delta_j})$

Experiments:

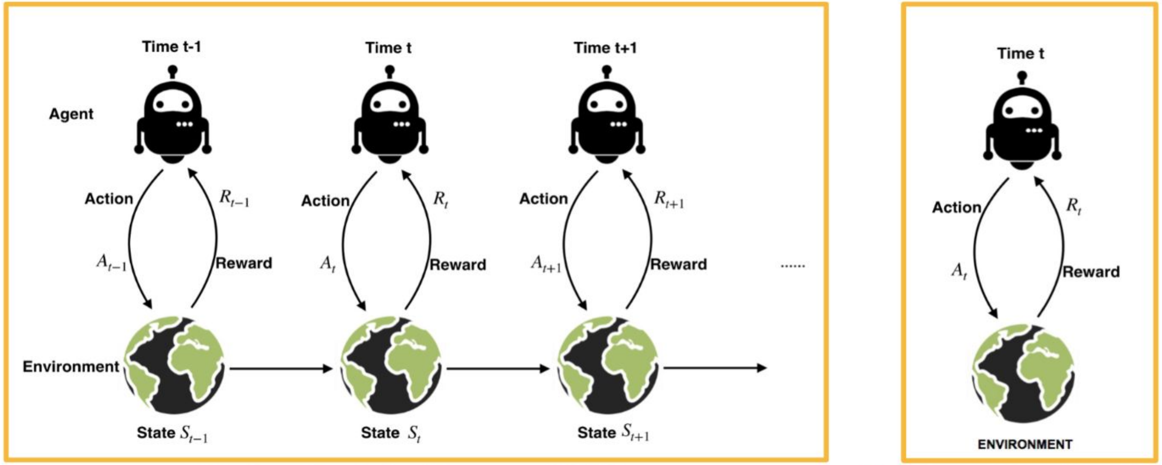




MDPs

RL vs Bandits

Markov Decision Processes (MDPs) provide a framework for modelling **sequential decision making**, where the environment has different states which change over time as a result of the agent's actions.



Markov Decision Process (MDP) Multi-arm Bandits (MAB)

- Bandit can be viewed as an MDP with one state and K actions.

Markov Decision Processes (MDPs)

$$[S], [A], H, \{\vec{P}\}_{[S]\times[A]\times[H]}, \{\mu\}_{[S]\times[A]\times[H]}, T, p_0$$

States

Actions

Time Horizon

Transition Probability

Mean reward

Episodes

Initial state

Assumptions:
Finite horizon
Time-dependent
Episodic

Policy $\pi = (\pi(\cdot,1), \pi(\cdot,2), \ldots, \pi(\cdot,H))$: a sequence of functions, where each $\pi(\cdot,t) : \mathcal{S} \to \mathcal{A}$ takes a state s as input and outputs an action a that will be taken in that round t

Regret can be expressed as

$$\mathcal{R}(T;M) = \sum_{k=1}^{T} \mathbb{E}\left[V_1^{\pi_*}(s_1^k) - V_1^{\pi_k}(s_1^k)\right]$$

Value function for the **used policy**

Value function for **optimal policy**

$s_1^k \sim p_0$

V_t^π : value function for policy π in round t

$$V_t^\pi(s) = \mathbb{E}\left[\sum_t^H \mu_{s_t,a_t,t} | \pi, s_t = s\right]$$

(Model-based) Exploration in MDPs

- Planning Phase: Compute policy and value function via backward inductions (using UCB or randomly sampled reward in TS)
- Sampling Phase: Act greedily according to the plan

Algorithm 4.6 Optimism-based RL

for episode $k = 1, 2, ..., K$ **do**
    for step $t = H, H-1, ..., 1$ **do**
        Construct UCB for each (s, a, t)
        Compute state-value function $Q_t^k(s, a)$
    end for
    for step $t = 1, 2, ..., H$ **do**
        Take action $a_t^k = \arg\max_a Q_t^k(s, a)$
    end for
end for

Algorithm 4.7 Posterior Sampling in RL

for episode $k = 1, 2, ..., K$ **do**
    for step $t = H, H-1, ..., 1$ **do**
        Sample from posterior distribution for each (s, a, t)
        Compute state-value function $Q_t^k(s, a)$
    end for
    for step $t = 1, 2, ..., H$ **do**
        Take action $a_t^k = \arg\max_a Q_t^k(s, a)$
    end for
end for

O-TS-MDP

Algorithm 1 O-TS-MDP

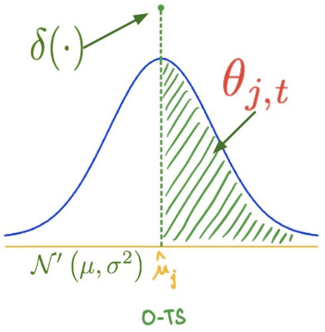
Input: MDP instance M, number of episodes T
Initialization:
Set $\widehat{O}_{s,a,t} \leftarrow 0$, $\widehat{P}_{s,a,t} \leftarrow \vec{0}$, $\widehat{\mu}_{s,a,t} \leftarrow 0, \forall (s,a,t)$
for episode $k = 1, 2, \dots, T$ **do**
  Set $\widetilde{V'}^{\pi_k}_{H+1} = \vec{0}$
  for $t = H, H-1, \dots, 1$ **do**
    for $s \in \mathcal{S}$ **do**
      for $a \in \mathcal{A}$ **do**
        Draw $\widetilde{\mu}_{s,a,t} \sim \mathcal{N}\left(\widehat{\mu}_{s,a,t}, \left(\sqrt{SH}\sigma^k_{s,a,t}\right)^2\right)$
        Set $\widetilde{\mu}'_{s,a,t} \leftarrow \max\{\widetilde{\mu}_{s,a,t}, \widehat{\mu}_{s,a,t}\}$
        Set $\widetilde{Q}_{s,a,t} \leftarrow \widetilde{\mu}'_{s,a,t} + \widehat{P}^{\intercal}_{s,a,t}\widetilde{V'}^{\pi_k}_{t+1}$
      end for
      Set $\pi_k(s,t) \leftarrow \arg\max_{a \in \mathcal{A}} \widetilde{Q}_{s,a,t}$
      Set $\widetilde{V'}^{\pi_k}_t(s) \leftarrow \widetilde{Q}_{s,\pi_k(s,t),t}$
    end for
  end for
  Sample $s^k_1 \sim p_0$, run π_k, and update $\widehat{\mu}_{s^k_t,\pi_k(s^k_t,t),t}$, $\widehat{O}_{s^k_t,\pi_k(s^k_t,t),t}$, and $\widehat{P}_{s^k_t,\pi_k(s^k_t,t),t}$ for all $t \in [H]$.
end for

$$\left(\sigma^k_{s,a,t}\right)^2 = \frac{\square \cdot H^2 \log(1/\delta)}{\widehat{O}_{s,a,t}}$$

Planning Phase

- Draw a sample from data dependent distribution
- Clip to non-negative value
- Compute Q value

Sampling Phase



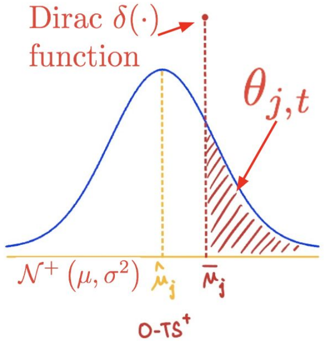
O-TS-MDP$^+$

$$\left(\sigma^k_{s,a,t}\right)^2 = \frac{\square \cdot H^2 \log(1/\delta)}{\widehat{O}_{s,a,t}}$$

Algorithm 2 O-TS-MDP$^+$

```
Input: MDP instance M, number of episodes T
Initialization:
  Set Ô_{s,a,t} ← 0, P̂_{s,a,t} ← 0⃗, μ̂_{s,a,t} ← 0, ∀(s,a,t)
for episode k = 1, 2, ..., T do
  Set Ṽ'^{π_k}_{H+1} = 0⃗
  for t = H, H − 1, ..., 1 do
    for s ∈ S do
      for a ∈ A do
        Draw μ̃_{s,a,t} ~ N(μ̂_{s,a,t}, (σ^k_{s,a,t})^2)
        Set μ̄_{s,a,t} ← μ̂_{s,a,t} + 2σ^k_{s,a,t}
        Set μ̃'_{s,a,t} ← max{μ̃_{s,a,t}, μ̄_{s,a,t}}
        Set Q̃_{s,a,t} ← μ̃'_{s,a,t} + P̂^T_{s,a,t} Ṽ'^{π_k}_{t+1}
      end for
      Set π_k(s,t) ← arg max_{a∈A} Q̃_{s,a,t}
      Set Ṽ'^{π_k}_t(s) ← Q̃_{s,π_k(s,t),t}
    end for
  end for
  Sample s^k_1 ~ p_0, run π_k, and update μ̂_{s^k_t,π_k(s^k_t,t),t},
  Ô_{s^k_t,π_k(s^k_t,t),t} and P̂_{s^k_t,π_k(s^k_t,t),t} for all t ∈ [H].
end for
```

- Draw a sample from data dependent distribution
- Compute Upper confidence bound
- Clip to UCB
- Compute Q value



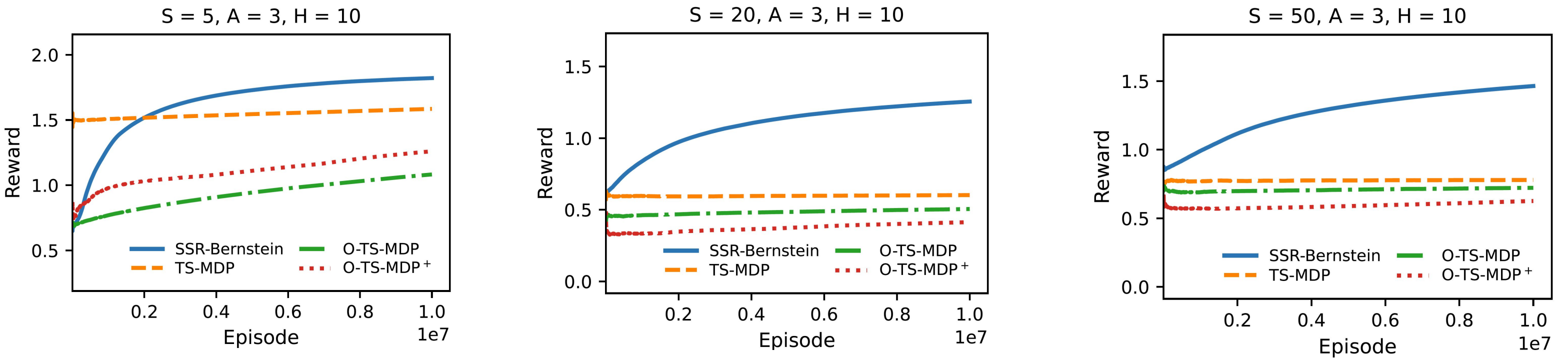
Experiments

Regret:

UCB-VI [Dann et al., 2017]	$\tilde{O}\left(\sqrt{ASH^3T}\right)$	Model-based	Deterministic
RLSVI [Russo, 2019]	$\tilde{O}\left(\sqrt{AS^2H^4T}\right)$	Model-free	Randomized
O-TS-MDP	$\tilde{O}\left(\sqrt{AS^2H^4T}\right)$	Model-based	Randomized
O-TS-MDP$^+$	$\tilde{O}\left(\sqrt{ASH^3T}\right)$	Model-based	Randomized

Near optimal

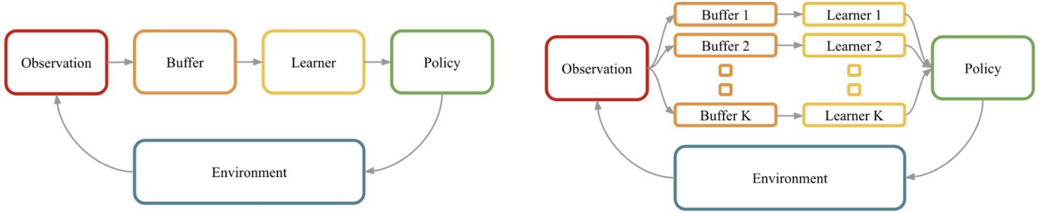
Experiments:



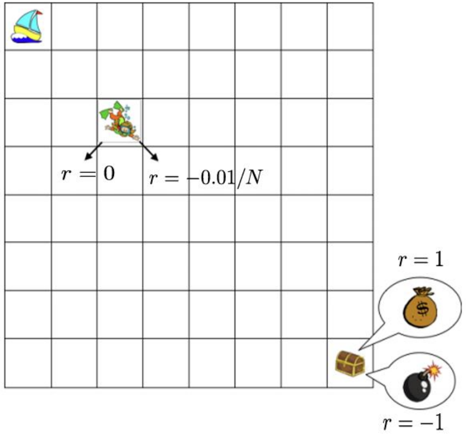
Shout out to Alan and Fred :)

Limitations and Future Work

- Better experiment/understanding
- Practical approaches
- Connection to differential privacy



(a) learning a single value function (b) learning multiple value functions in parallel



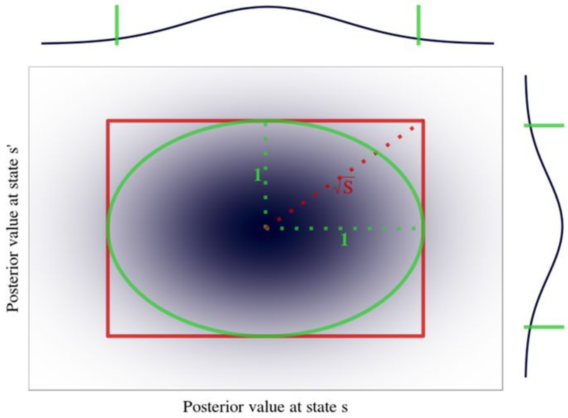
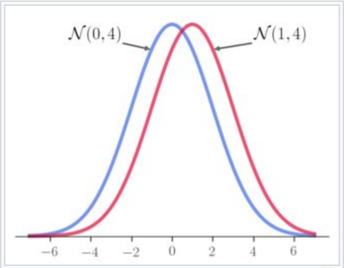


Figure 3. Union bounds give loose rectangular confidence sets.

Gaussian Mechanism [edit]

Analogous to Laplace mechanism, Gaussian mechanism adds noise drawn from a Gaussian distribution whose variance is calibrated according to the sensitivity and privacy parameters. For any $\delta \in (0,1)$ and $\epsilon \in (0,1)$, the mechanism defined by:



Gaussian mechanism.

$$\mathcal{M}_{\text{Gauss}}(x, f, \epsilon, \delta) = f(x) + \mathcal{N}\left(\mu = 0, \sigma^2 = \frac{2\ln(1.25/\delta)\cdot(\Delta f)^2}{\epsilon^2}\right)$$

provides (ϵ, δ)-differential privacy.

Take Away

- We draw inspiration from TS and UCB and analyze optimistic Thompson sampling (O-TS) and O-TS$^+$ in bandit.
- We propose O-TS-MDP, a computationally efficient and theoretically elegant model-based learning algorithm for episodic MDPs.
- We also propose O-TS-MDP$^+$ which achieves the (near)-optimal regret bound with a more aggressive clipping strategy of the posterior distribution.
- The key of our algorithms is to uses optimistic clipping of Gaussian distribution to model the reward distributions and drive exploration.

Thanks for listening!

And happy to hear any questions and feedbacks :)