# Multi-Armed Bandits

Yuhang Wu

B9145: Topics in Trustworthy AI

April 10, 2025

# The Bandit

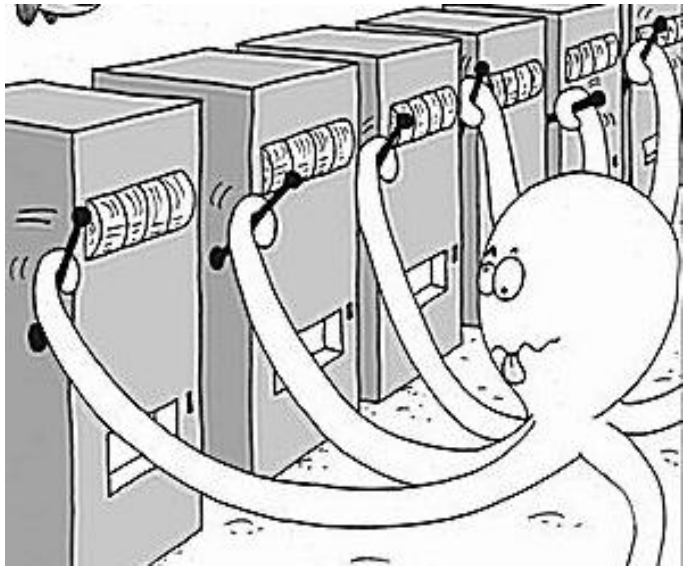


Figure: Multi-Armed Bandit.

| Time | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| Arm 1 | \$1 | \$0 | | | \$1 | \$1 | \$0 | | |
| Arm 2 | | | \$1 | \$0 | | | | | |

Which arm to pull next?

# Motivation

Many real-world problems can be modeled as multi-armed bandit problems:

- Clinical trials
- Online advertising
- Recommender systems

⋮

# Mathematical Formulation

Consider a bandit problem with $K$ arms. At each time step $t = 1, 2, ...,$

- The agent selects an arm $A_t \in \{1, 2, ..., K\}$.
- The agent receives a reward $X_t \sim P_{A_t}$ from the selected arm, where $P_k$ is the distribution of rewards for arm $k$.

**Remark:**

- $P_1, ..., P_k$ are unknown to the agent. For simplicity, we assume that $P_k$ is stationary.
- Common distributional assumptions are: Bernoulli, Gaussian, subgaussian, supported on $[0, 1]$, etc.
- We usually care about the expected reward, i.e., $\mu_k = \mathbb{E}[X_t | A_t = k]$ is the expected value of distribution $P_k$.

# Objective

Want to maximize

$$\sum_{t=1}^{T} \mathbb{E}[X_t],$$

where the expectation is taken over the randomness of the arm selection (policy) and the reward distributions. Equivalently, want to minimize the **regret**:

$$R(T) = \sum_{t=1}^{T} \mu^* - \mathbb{E}[X_t] = T\mu^* - \sum_{t=1}^{T} \mathbb{E}[X_t],$$

where $\mu^* = \max_k \mu_k$ is the expected reward of the best arm. A low bar for the regret is to achieve $R(T) = o(T)$.

## Lower Bound

Minimax: $R(T) = \Omega(\sqrt{KT})$.

# Exploration vs. Exploitation

At each time step, we can either:

- **Explore**: Select an arm that we have not tried enough yet to gather more information about its reward distribution.
- **Exploit**: Select the arm that we believe has the highest expected reward based on the information we have gathered so far.

**Trade-off:**

- If we explore too much, we may pull suboptimal arms for too many times and incur high regret.
- If we exploit too much, we may miss out on better arms and incur high regret.

**Goal**

Try various actions while progressively favor high-reward actions.

# Explore-Then-Commit

**Algorithm:**

- Pull each arm $m$ times (assume $mK < T$).
- Compute the empirical mean of each arm $\hat{\mu}_k$.
- For $t = mK + 1, ..., T$, select the arm with the highest empirical mean (ties broken arbitrarily).

**Analysis:**

- Larger $m$ means more exploration.
- Suppose $K = 2$. If $T$ is known in advance, under some conditions we can choose $m$ to obtain $R(T) = O(T^{2/3})$.
- If optimality gap is also known, can obtain $R(T) = O(\sqrt{T})$.

# Explore-Then-Commit

**Problems:**

- Really is a "greedy" algorithm.
- Need to know $T$ in advance, not adaptive.
- Not optimal if optimality gap is unknown.
- Better approaches exist, but Explore-Then-Commit is often a good place to start when analyzing a bandit problem.

**Takeaway**

We should explore adaptively.

# Epsilon-Greedy

**Algorithm:**

- Input $\epsilon \in (0, 1)$.
- For $t = 1, 2, ...$:
  - With probability $\epsilon$, select a random arm $A_t \sim \text{Uniform}(1, 2, ..., K)$.
  - With probability $1 - \epsilon$, select the arm with the largest $\hat{\mu}_t(k)$.
  - Pull arm $A_t$ and observe reward $X_t$.
  - Update $\hat{\mu}_t(k)$.
- This is really easy to implement!
- Intuition clear! Trade-off controlled by a single parameter $\epsilon$.

# Epsilon-Greedy

**Analysis:**

- We're always exploring, so $R(T)$ is linear.
- Tuning $\epsilon$ is nontrivial.

So, let's decay $\epsilon$ over time. Choose

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

where $d$ is the smallest optimality gap and $C$ is a large constant. Then, we can show that roughly $R(T) = O\left(\frac{K}{d}\log T\right)$ (this is instance-dependent optimal).

# Epsilon-Greedy

**Problems:**

- Need to know $d$ in advance.
- If $d$ is small, the regret constant is large.

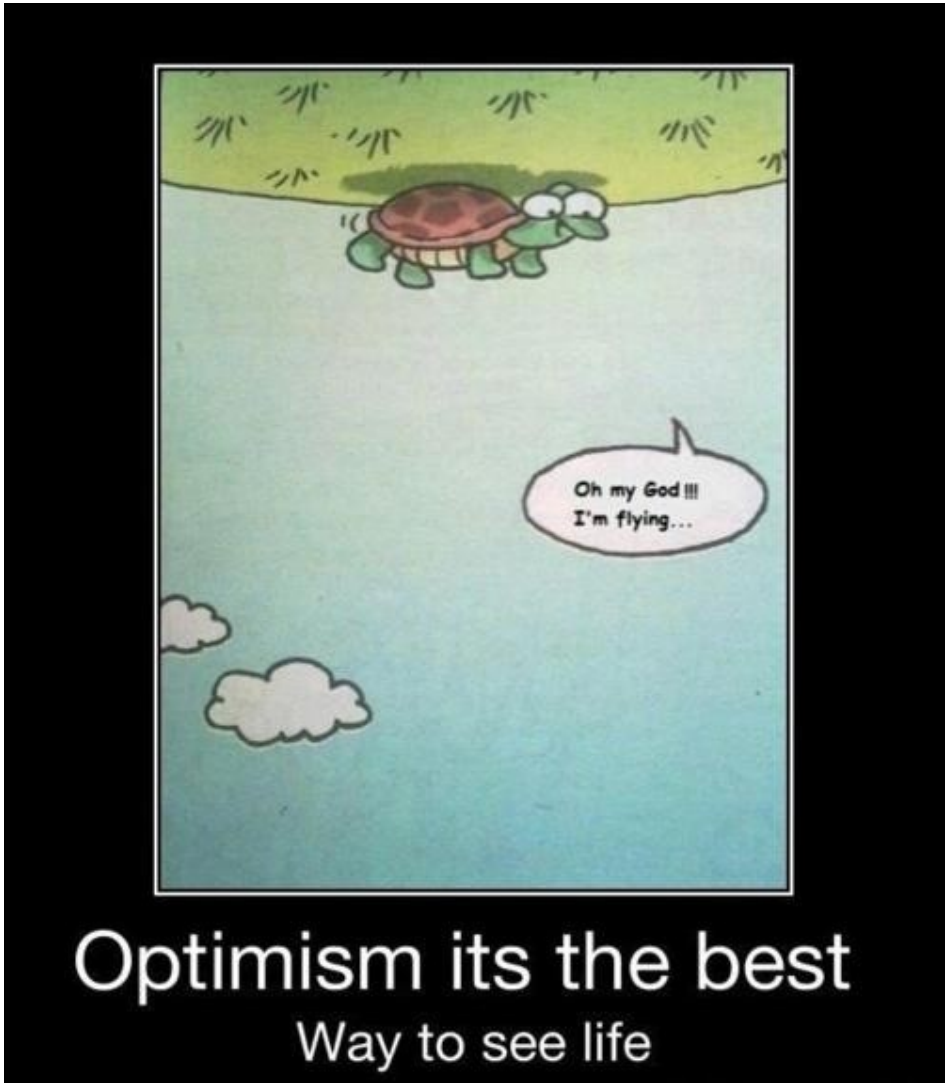
Intuitively, the exploration is inefficient because we allocate exploration effort equally, even if empirical expected rewards may be very different across arms.

**Takeaway**

We should use past information to inform our exploration.

The more uncertain we are about an action's reward, the more important it is to explore that action, as it could turn out to be the best action. But how to quantify uncertainty?

# Optimism in the Face of Uncertainty



# Upper Confidence Bound (UCB)

**Idea/Algorithm:**

- Construct an upper-confidence $U_t(k)$ for each arm $k$ s.t.

$$\mu_k \leq \hat{\mu}_t(k) + U_t(k) \text{ with high probability.}$$

- $U_t(k)$ bounds the uncertainty and depends on $N_t(k)$:

$$\text{larger } N_t(k) \implies \text{smaller } U_t(k).$$

- Select the arm with the largest upper confidence bound:

$$A_t = \underset{k}{\operatorname{argmax}} \left\{ \hat{\mu}_t(k) + U_t(k) \right\}.$$

$\hat{\mu}_t(k)$ represents the exploitation and $U_t(k)$ represents the exploration.

# Upper Confidence Bound (UCB)

**Analysis:**

- If $P_k$'s are 1-subgaussian, one choice for $U_t(k)$ is:

$$U_t(k) = \sqrt{\frac{2\log(1/\delta)}{N_t(k)}} \quad \text{for } \delta \in (0,1),$$

because (abuse of notation)

$$\mathbb{P}\left(\mu_k > \frac{1}{n}\sum_{t=1}^{n} X_t + \sqrt{\frac{2\log(1/\delta)}{n}}\right) \leq \delta.$$

- If we choose $\delta = \frac{1}{1+t\log^2(t)}$, can show that $R(T) = \tilde{O}(\sqrt{KT})$, which is minimax optimal (can also show instance-dependent optimality).

# Upper Confidence Bound (UCB)

- **Intuition**: say $k = 1$ is the optimal arm. For large $t$, with high probability,

$$\hat{\mu}_t(k) + U_t(k) \leq \mu_1 \leq \hat{\mu}_t(1) + U_t(1) \text{ for all } k \neq 1.$$

- Other distributional assumption + similar concentration inequalities lead to similar results (e.g., bounded support + Hoeffding).
- In practice, can start by pulling each arm once.

**Takeaway**

Using past information to inform our exploration is a good idea.

What else can we do?

# Thompson Sampling

Let's put on our Bayesian hats. Consider the Beta-Bernoulli bandit:

- $P_k = \text{Ber}(\theta_k)$.
- Prior $\theta_k \sim \text{Beta}(\alpha_k, \beta_k)$ (if $\alpha_k = \beta_k = 1$, we have $U[0,1]$).
- If we choose arm $k$, we observe $X_t \in \{0,1\}$ and update the posterior:

$$\theta_k \sim \text{Beta}(\alpha_k + X_t, \beta_k + 1 - X_t).$$

At any time $t$, the posterior distribution of $\theta_k$ captures all the information we have about arm $k$, including the expected reward and uncertainty. This leads to an algorithm.

# Thompson Sampling

**Algorithm:**

- Initialize $\alpha_0(k), \beta_0(k)$ for all arms $k$.
- For $t = 0, 1, 2, ...$
  - Sample $\hat{\theta}_t(k) \sim \text{Beta}(\alpha_t(k), \beta_t(k))$ for all arms $k$.
  - Pull arm $A_t = \text{argmax}_k\{\hat{\theta}_t(k)\}$ and observe reward $X_t$
  - Update the posterior:

$$\alpha_{t+1}(A_t) = \alpha_t(A_t) + X_t, \quad \beta_{t+1}(A_t) = \beta_t(A_t) + 1 - X_t.$$

  - For all $k \neq A_t$, $\alpha_{t+1}(k) = \alpha_t(k)$ and $\beta_{t+1}(k) = \beta_t(k)$.

# Thompson Sampling

**General Algorithm:**

- Initialize prior for each arm $k$.
- For $t = 0, 1, 2, ...$
  - Sample $\hat{\theta}_t(k)$ from the posterior distribution of arm $k$.
  - For each arm, calculate the expected reward given the sampled $\hat{\theta}_t(k)$.
  - Pull the arm with the highest expected reward and observe reward $X_t$.
  - Update the posterior distribution of the selected arm based on $X_t$.
  - For all other arms, keep the posterior distribution unchanged.

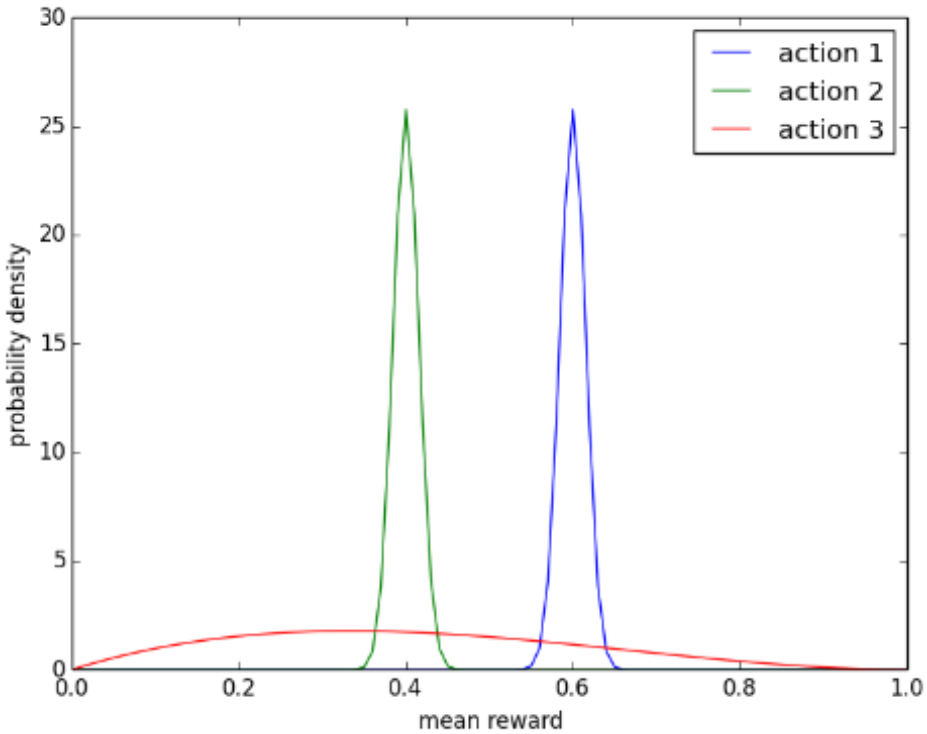
# Thompson Sampling

**Analysis:**

- The Bayesian regret $BR(T)$ is defined as the expected $R(T)$ over the prior.
- Under some conditions, for Beta-Bernoulli bandits can show that $BR(T) = O(\sqrt{KT \log T})$ (optimal up to log factors).
- Can consider many other specifications. E.g.,
  - Gaussian-Gaussian bandit: $P_k$ normal with known variance, unknown mean follows normal prior.
  - Log-Gaussian: $P_k$ log-normal with known variance, unknown mean follows log-normal prior.
  - $P_k$ is in 1-dim exponential family with Jeffreys prior.
- Usually, Thompson Sampling is at least near-optimal.
- When posterior sampling is intractable, use numerical methods like Gibbs sampling, Langevin Monte Carlo, etc.

# Thompson Sampling

Why does it work? Back to Beta-Bernoulli bandits.




Beta distributions with
$(\alpha_1, \beta_1) = (601, 401)$,
$(\alpha_2, \beta_2) = (401, 601)$,
$(\alpha_3, \beta_3) = (2, 3)$.

Prob of pulling the arms are:
0.82, 0, 0.18 $\Longleftrightarrow$ prob that the random estimate drawn for the action exceeds those drawn for others.

- Discard the arm for which we're quite confident that it is not the best.
- Encourage exploration.

# Thompson Sampling

**Intuition.** Two equivalent ways to view Thompson Sampling:

- We select an arm according to the posterior probability that the arm is optimal.
- We sample an environment from the posterior and play the optimal action in that environment.

Formally,

$$\mathbb{P}(A^* = \cdot \,|\mathcal{F}_{t-1}) = \mathbb{P}(A_t = \cdot \,|\mathcal{F}_{t-1}),$$

i.e., the conditional distribution of the optimal arm given the history is the same as the conditional distribution of the selected arm given the history.

**Takeaway**

By sampling actions according to the posterior probability that they are optimal, we continue to sample all actions that could plausibly be optimal, while shifting sampling away from those that are unlikely to be optimal.

# Contextual Bandits

In many problems, we may have additional information that can help us predict the rewards of the arms. Consider personalized recommendation:

- User features may include historical activities, demographics.
- Content features may include genre, length, etc.

We can call the combination of user and content features the context.

Exploration-exploitation trade-off $\Longrightarrow$ maximizing user satisfaction in the long run v.s. gathering information about the goodness of match between user and content.

# Contextual Bandits

**Formulation:**

- Let $C$ denote the set of contexts.
- At time $t$, the agent observes context $c_t \in C$, selects an arm $A_t \in \{1, 2, ..., K\}$, and receives a reward $X_t = r(c_t, k) + \eta_t$, where $r(c_t, k)$ is the (deterministic) reward function and $\eta_t$ is the noise.

The regret is defined as:

$$R(T) = \mathbb{E}\left[\sum_{t=1}^{T} \max_k r(c_t, k) - \sum_{t=1}^{T} X_t\right].$$

Naively, we can set an arm for each context-action pair. However, $|C|$ can be large. So, need some structure.

# Linear Bandits

Suppose we have a feature map $\psi(c, a) \in \mathbb{R}^d$, and for some unknown $\theta^* \in \mathbb{R}^d$, we have:

$$r(c, a) = \langle \theta^*, \psi(c, a) \rangle, \quad \text{for all } c, a.$$

Equivalently, at time $t$, we can define action set $\mathcal{A}_t \subseteq \mathbb{R}^d$ to be

$$\mathcal{A}_t = \{\psi(c_t, k) : k = 1, 2, ..., K\}.$$

Then, the reward is given by $r(A_t) = \langle \theta^*, A_t \rangle$ for $A_t \in \mathcal{A}_t$. The regret is

$$R(T) = \mathbb{E}\left[\sum_{t=1}^{T} \max_{a \in \mathcal{A}_t} \langle \theta^*, a \rangle - \sum_{t=1}^{T} X_t\right],$$

i.e., all that matters is the feature vector that results from choosing a given action and not the action itself.

# Linear Bandits

What algorithms can we use?

- $\epsilon$-greedy? Not efficient, especially if $\mathcal{A}_t$ is large. $\Longrightarrow$ often used as a baseline.
- UCB $\Longrightarrow$ LinUCB.
- Thompson Sampling $\Longrightarrow$ LinTS.

## Lower Bound

Usually, with some assumptions on $\mathcal{A}_t$,

$$R(T) = \Omega\left(d\sqrt{T}\right).$$

Before, with $K$ arms, we had $R(T) = \Omega(\sqrt{KT})$.

# LinUCB

To extend UCB to linear bandits, we need to be able to estimate $\theta^*$ and construct a confidence set for it. One way is to use Ridge regression:

$$\hat{\theta}_t = \operatorname*{argmin}_{\theta} \sum_{s=1}^{t-1} (X_s - \langle \theta, A_s \rangle)^2 + \lambda \|\theta\|_2^2 ,$$

where $\lambda > 0$ is a regularization parameter. The solution is:

$$\hat{\theta}_t = V_t^{-1} \sum_{s=1}^{t-1} X_s A_s,$$

where

$$V_0 = \lambda I, \quad V_t = V_0 + \sum_{s=1}^{t-1} A_s A_s^T .$$

# LinUCB

The following holds with high probability:

$$\theta^* \in \mathcal{E}_t = \left\{ \theta \in \mathbb{R}^d : ||\theta - \hat{\theta}_t||_{V_t}^2 \leq \beta_t \right\}$$

for some increasing sequence $\beta_t$ and $||x||_A$ for $A \succ 0$ is $\sqrt{x^T A x}$. Using $\mathcal{E}_t$ as the confidence set, the algorithm goes by solving

$$(A_t, \cdot) = \underset{(a,\theta) \in \mathcal{A}_t \times \mathcal{E}_t}{\operatorname{argmax}} \langle \theta, a \rangle .$$

If $\mathcal{A}_t$ is finite, can solve

$$A_t = \underset{a \in \mathcal{A}_t}{\operatorname{argmax}} \left\{ \left\langle \hat{\theta}_t, a \right\rangle + \underbrace{\sqrt{\beta_t} ||a||_{V_t^{-1}}}_{\text{like } U_t} \right\} .$$

Can show $R(T) = \tilde{O}(d\sqrt{T})$.

# LinUCB

**Takeaway:**

- Extending UCB to LinUCB is like extending mean estimation to linear regression.
- The (technically) hard part is to construct the confidence set $\mathcal{E}_t$.
- Depending on the structure of $\mathcal{A}_t$, the bilinear optimization may also be hard.

Now comes LinTS...

# LinTS

The procedure is the same as before. All that's different is that the prior and the posterior are now multivariate.

- Choose a prior for $\theta^*$.
- For $t = 1, 2, ...$
  - Sample $\theta_t$ from the posterior.
  - Choose $A_t = \text{argmax}_{a \in \mathcal{A}_t} \langle \theta_t, a \rangle$ and observe reward $X_t$.
  - Update the posterior of $\theta^*$ based on $X_t$.

Under some conditions, can show that $BR(T) = \tilde{O}(d\sqrt{T})$.

# LinTS

**Discussion:**

- Provided that we can do posterior sampling, the optimization problem is very simple compared to LinUCB.
- For more abstract settings (like texts), it's not often clear what a confidence set looks like. In that sense, Thompson Sampling is more flexible than the optimism principle.
- Empirically, Thompson Sampling is also quite strong.

**Takeaway**

Keep the Bayesian hat on?

# References


- Chapelle, O., & Li, L. (2011). An empirical evaluation of thompson sampling. Advances in neural information processing systems, 24.
- Korda, N., Kaufmann, E., & Munos, R. (2013). Thompson sampling for 1-dimensional exponential family bandits. Advances in neural information processing systems, 26.
- Lattimore, T., & Szepesvári, C. (2020). Bandit algorithms. Cambridge University Press.
- Li, L., Chu, W., Langford, J., & Schapire, R. E. (2010, April). A contextual-bandit approach to personalized news article recommendation. In Proceedings of the 19th international conference on World wide web (pp. 661-670).
- Russo, D. J., Van Roy, B., Kazerouni, A., Osband, I., & Wen, Z. (2018). A tutorial on thompson sampling. Foundations and Trends® in Machine Learning, 11(1), 1-96.