Lecture 6

Bandits, Monte Carlo Prediction, and Control

Goals of this Lecture

1. Define regret and evaluate exploration strategies including ε-greedy, UCB, and Thompson Sampling.
2. Learn how to estimate value functions via Monte Carlo methods using sampled trajectories: first-visit and every-visit estimators, and their incremental updates.
3. Develop model-free policy improvement techniques using Monte Carlo control and GLIE strategies: Exploring Starts, ε-soft policies.

6.1 Recap: Multi-Armed Bandits

Setup. In the *multi-armed bandit* problem, the agent repeatedly chooses from a finite set of actions (arms) $\mathcal{A} = \{1, \ldots, K\}$, receiving a stochastic reward each time. The reward distribution for each arm is unknown.

Goal. The agent aims to maximize cumulative reward over time, which requires balancing:

- **Exploration:** trying different arms to learn their rewards,
- **Exploitation:** choosing arms believed to be optimal.

Performance Metric. Let $q(a)$ be the expected reward of arm a, and $v^* = \max_a q(a)$. The cumulative *regret* after T rounds is:

$$\text{Regret}(T) := Tv^* - \mathbb{E}\left[\sum_{t=1}^{T} q(A_t)\right],$$

which quantifies how much reward was lost compared to always playing the best arm.

Key Insight. Regret arises from pulling suboptimal arms. The agent must learn to identify the best arms quickly, minimizing regret by allocating actions wisely over time.

6.2 Exploration Strategies

We now describe four foundational strategies for managing the exploration–exploitation tradeoff in multi-armed bandit problems: *Explore-then-Exploit*, ε-greedy, Upper Confidence Bound (UCB), and Thompson Sampling.

Empirical Action-Value Estimates. All of the methods we discuss rely on estimates of the expected reward for each arm $a \in \mathcal{A}$. We denote by $\hat{q}_a(t)$ the empirical average reward observed from arm a up to time t:

$$\hat{q}_a(t) := \frac{1}{N_a(t)} \sum_{k=1}^{t-1} \mathbb{1}\{A_k = a\} \cdot R_k,$$

where $N_a(t)$ is the number of times arm a has been selected in the first $t-1$ rounds. If $N_a(t) = 0$, we define $\hat{q}_a(t) := 0$ or initialize it arbitrarily to ensure proper exploration.

Explore-then-Exploit. One of the simplest strategies is to divide learning into two phases:

- **Exploration phase:** For the first T_0 rounds, sample each arm a fixed number of times (e.g., uniformly at random),
- **Exploitation phase:** For the remaining rounds, repeatedly select the arm with the highest estimated reward:

$$A_t = \arg\max_{a \in \mathcal{A}} \hat{q}_a(T_0).$$

This approach guarantees that all arms are explored initially, after which the best one is exploited. While easy to implement, it can perform poorly if T_0 is not chosen carefully—too little exploration risks missing the best arm, while too much incurs unnecessary regret.

The following strategies address this shortcoming by interleaving exploration and exploitation throughout learning.

ε-Greedy. The ε-greedy strategy balances exploration and exploitation using a simple randomized rule. At each round t, the agent selects:

- With probability $1 - \varepsilon$: the arm with the highest empirical estimate:

$$A_t = \arg\max_{a \in \mathcal{A}} \hat{q}_a(t),$$

- With probability ε: an arm chosen uniformly at random from $\mathcal{A}$.

The exploration parameter $\varepsilon \in (0, 1)$ determines how frequently the agent explores. A small ε leads to more exploitation, while a larger value promotes exploration. To improve long-term performance, it is common to decay ε over time, for example using $\varepsilon_t = 1/\sqrt{t}$, to explore more in early rounds and exploit later.

Upper Confidence Bound (UCB). UCB is a frequentist algorithm based on the principle of optimism in the face of uncertainty. It augments the empirical estimate $\hat{q}_a(t)$ with a confidence bonus that decreases with $N_a(t)$, the number of times arm a has been selected:

$$A_t = \arg\max_{a \in \mathcal{A}} \left[\hat{q}_a(t) + c \cdot \sqrt{\frac{\log t}{N_a(t)}} \right],$$

where $c > 0$ is a tunable parameter controlling the degree of optimism. The $\log t/N_a(t)$ term encourages early exploration of all arms and gradually shifts toward exploitation as uncertainty decreases.

Thompson Sampling. Thompson Sampling is a Bayesian exploration strategy that maintains a posterior distribution over each arm's expected reward. At each round t, the agent proceeds as follows:

1. For each arm $a \in \mathcal{A}$, sample $\theta_a(t)$ from the posterior distribution over $q(a)$,
2. Select the arm with the highest sampled value:

$$A_t = \arg\max_{a \in \mathcal{A}} \theta_a(t).$$

The sampled value $\theta_a(t)$ represents a plausible estimate of the expected reward $q(a)$, drawn from the agent's current belief. This randomization induces structured exploration: arms with greater uncertainty (i.e., wider posterior distributions) are more likely to be selected, even if their current empirical mean is lower.

Example: Bernoulli Bandits. In the Bernoulli bandit setting, each arm yields binary rewards $R_t \in \{0, 1\}$, with unknown mean $q(a) = \mathbb{E}[R_t \mid A_t = a]$. The agent models its uncertainty about each arm's mean reward $q(a)$ using a Bayesian approach. Specifically, the agent treats $q(a)$ as a random variable with a prior distribution:

$$q(a) \sim \text{Beta}(\alpha_a, \beta_a),$$

where $\alpha_a > 0$ and $\beta_a > 0$ are parameters that encode the agent's belief about how likely the arm is to yield rewards of 1 or 0, respectively.

- The Beta distribution is a natural prior for Bernoulli outcomes because it is the *conjugate prior*—the posterior remains a Beta distribution after observing data.
- Initially, we use the uniform prior $\text{Beta}(1, 1)$, which represents no prior preference for 0 or 1 outcomes.
- Each time arm a is pulled and a reward $R_t \in \{0, 1\}$ is observed, the agent updates the posterior parameters:

$$\alpha_a \leftarrow \alpha_a + R_t, \qquad \beta_a \leftarrow \beta_a + (1 - R_t).$$

  That is:
  - If $R_t = 1$, we increment α_a, reinforcing the belief that arm a tends to yield reward 1,
  - If $R_t = 0$, we increment β_a, reinforcing the belief that arm a tends to yield reward 0.
- After the update, the posterior over $q(a)$ becomes $\text{Beta}(\alpha_a, \beta_a)$. To choose the next action, we:
  1. Sample a mean reward estimate:

$$\theta_a(t) \sim \text{Beta}(\alpha_a, \beta_a),$$

  2. Select the arm with the highest sample:

$$A_t = \arg\max_{a \in \mathcal{A}} \theta_a(t).$$

Interpretation. This strategy is Bayesian because it explicitly models and updates the agent's uncertainty about each arm's mean reward via a posterior distribution. The parameters α_a and β_a track the number of observed rewards and failures, respectively. Early in training, the posteriors are wide (uncertain), promoting exploration. Over time, the posterior concentrates around the empirical mean, leading to exploitation. Sampling from the posterior thus naturally balances exploration and exploitation without requiring explicit tuning of exploration parameters.

6.3 Regret Bounds and Performance Guarantees

We now examine the performance of different exploration strategies in terms of cumulative regret. Recall:

$$\text{Regret}(T) := T \cdot q^* - \sum_{t=1}^{T} \mathbb{E}[q(A_t)],$$

where $q^* = \max_a q(a)$ is the mean reward of the best arm.

6.3.1 Regret of Explore-then-Exploit.

Assume we explore uniformly for T_0 rounds, selecting each arm approximately T_0/K times. Let $\hat{q}_a$ be the empirical mean reward of arm a after this phase, and let $\hat{a} = \arg\max_a \hat{q}_a$ be the empirically best arm. In the remaining $T - T_0$ rounds, the agent exploits arm $\hat{a}$.

If T_0 is large enough to accurately estimate the means (e.g., $T_0 = \Theta(K \log T)$), the probability of selecting a suboptimal arm during exploitation is small. The regret decomposes as:

$$\text{Regret}(T) \leq T_0 \cdot \Delta_{\max} + (T - T_0) \cdot \mathbb{P}[\hat{a} \neq a^*] \cdot \Delta_{\max},$$

where a^* is the true optimal arm and $\Delta_{\max} := \max_a(\mu^* - \mu_a)$. Using concentration bounds, it can be shown that with $T_0 = O\left(\frac{K}{\Delta^2} \log T\right)$, the total regret is:

$$\text{Regret}(T) = O\left(\frac{K}{\Delta} \log T\right),$$

where $\Delta = \min_{a \neq a^*} \Delta_a$ is the smallest suboptimality gap. While this matches the asymptotic rate of UCB and Thompson Sampling in the gap-dependent case (assuming knowledge of Δ), the constants are worse and the method lacks adaptivity.

Theorem 6.1 (Regret of Explore-then-Exploit)**.** *Let $\mathcal{A} = \{1, \ldots, K\}$ be a finite set of arms, and assume rewards are bounded in $[0, 1]$. Let $q(a) := \mathbb{E}[R_t \mid A_t = a]$ denote the expected reward of arm a, and define the optimal value $q^* := \max_a q(a)$. Let the suboptimality gap be $\Delta_a := q^* - q(a)$, and define $\Delta := \min_{a \neq a^*} \Delta_a$, where $a^* := \arg\max_a q(a)$ is the optimal arm.*

Suppose the agent explores uniformly for T_0 rounds, sampling each arm approximately T_0/K times, and then exploits the empirically best arm $\hat{a} := \arg\max_a \hat{q}_a$ for the remaining $T - T_0$ rounds. Then, for

$$T_0 = \left\lceil \frac{2K}{\Delta^2} \log T \right\rceil,$$

the expected regret satisfies:

$$\textit{Regret}(T) = O\left(\frac{K}{\Delta} \log T\right).$$

Proof. We decompose the regret as:

$$\text{Regret}(T) = \underbrace{T_0 \cdot \Delta_{\max}}_{\text{exploration}} + \underbrace{(T - T_0) \cdot \mathbb{P}[\hat{a} \neq a^*] \cdot \Delta_{\max}}_{\text{exploitation}},$$

where $\Delta_{\max} := \max_{a \neq a^*} \Delta_a$.

Let $n := T_0/K$ denote the number of samples per arm during exploration. Using Hoeffding's inequality, we have for each arm a,

$$\mathbb{P}[\,|\hat{q}_a - q(a)| > \epsilon\,] \leq 2\exp(-2n\epsilon^2).$$

To misidentify the best arm, there must exist an arm $a \neq a^*$ such that:

$$\hat{q}_a > \hat{q}_{a^*} \quad \Rightarrow \quad (\hat{q}_a - q(a)) + (q(a^*) - \hat{q}_{a^*}) > \Delta_a$$

Further by noting that the last condition requires,

$$(\hat{q}_a - q(a)) > \frac{\Delta_a}{2} \quad \text{or} \quad (q(a^*) - \hat{q}_{a^*}) > \frac{\Delta_a}{2}$$

we can bound $\mathbb{P}[\hat{a} \neq a^*]$ using the union bound:

$$\mathbb{P}[\hat{a} \neq a^*] \leq \sum_{a \neq a^*} \left[\mathbb{P}\left(\hat{q}_a > q(a) + \frac{\Delta_a}{2} \right) + \mathbb{P}\left(\hat{q}_{a^*} < q(a^*) - \frac{\Delta_a}{2} \right) \right].$$

Each term is bounded by Hoeffding's inequality:

$$\mathbb{P}[\hat{a} \neq a^*] \leq 2(K-1)\exp\left(-\frac{n\Delta^2}{2} \right).$$

Choosing $n = \frac{2}{\Delta^2}\log T$ (so $T_0 = \frac{2K}{\Delta^2}\log T$), we get:

$$\mathbb{P}[\hat{a} \neq a^*] \leq \frac{2(K-1)}{T}.$$

Substituting back into the regret expression:

$$\text{Regret}(T) \leq T_0 \cdot \Delta_{\max} + T \cdot \frac{2(K-1)}{T} \cdot \Delta_{\max} = O\left(\frac{K}{\Delta^2} \log T \cdot \Delta_{\max} \right).$$

Since $\Delta_{\max} \leq 1$, and assuming $\Delta_{\max} = O(\Delta)$, we obtain:

$$\text{Regret}(T) = O\left(\frac{K}{\Delta} \log T \right).$$

□

6.3.2 Regret of ε-Greedy.

With a fixed $\varepsilon > 0$, exploration occurs at a constant rate, leading to linear regret:

$$\text{Regret}(T) = O(\varepsilon T + \log T).$$

To achieve sublinear regret, ε must decay over time. For example, $\varepsilon_t = \Theta(1/\sqrt{t})$ yields regret $O(\sqrt{T \log T})$, though this requires careful tuning.

Theorem 6.2 (Regret of ε-Greedy)**.** *Consider the multi-armed bandit problem with K arms, each having a fixed mean reward $q(a)$, and let $a^* := \arg\max_a q(a)$ denote the optimal arm. Suppose the agent follows the ε-greedy strategy with:*

- *either a fixed exploration parameter $\varepsilon \in (0, 1)$,*
- *or a time-dependent schedule $\varepsilon_t = c/\sqrt{t}$ for some $c > 0$.*

Then:

1. *With fixed ε, the expected cumulative regret after T rounds satisfies:*

$$\textit{Regret}(T) = O(\varepsilon T + \log T).$$

2. *With decaying $\varepsilon_t = \Theta(1/\sqrt{t})$, the expected cumulative regret satisfies:*

$$\textit{Regret}(T) = O\left(\sqrt{T \log T}\right).$$

Proof. We decompose the expected regret into two parts: regret due to exploration, and regret due to incorrect exploitation.

(1) Exploration Regret. At each round t, with probability ε, the agent selects an arm uniformly at random. Each arm is selected with probability ε/K, including suboptimal ones. So, for each suboptimal arm $a \neq a^*$, the expected number of times it is pulled due to exploration is:

$$\mathbb{E}[N_a^{\text{explore}}(T)] = \frac{\varepsilon}{K} T.$$

Hence, the total exploration regret is:

$$\sum_{a \neq a^*} \Delta_a \cdot \mathbb{E}[N_a^{\text{explore}}(T)] \leq \varepsilon T \cdot \Delta_{\max}.$$

(2) Exploitation Regret. With probability $1 - \varepsilon$, the agent selects the arm with the highest empirical mean. Let $N_a^{\text{exploit}}(T)$ denote the number of times suboptimal arm a is chosen during exploitation. For arm $a \neq a^*$, we bound this using Hoeffding's inequality.

Let $\hat{q}_a(t)$ be the empirical mean of arm a after n pulls. By Hoeffding's inequality, for any $\delta > 0$, we have:

$$\mathbb{P}\left(|\hat{q}_a(t) - q(a)| > \delta\right) \leq 2\exp(-2\delta^2 n).$$

To confuse a suboptimal arm a with the optimal one, we must have:

$$\hat{q}_a(t) \geq \hat{q}_{a^*}(t) \quad \Rightarrow \quad \hat{q}_a(t) \geq q(a) + \frac{\Delta_a}{2}, \quad \text{or} \quad \hat{q}_{a^*}(t) \leq q(a^*) - \frac{\Delta_a}{2}.$$

Each of these events happens with probability at most $\exp(-\Delta_a^2 n/2)$, so the probability of incorrect selection at round t is at most:

$$2\exp(-\Delta_a^2 n/2).$$

Thus, summing over time, the expected number of times arm a is selected due to mistaken exploitation is bounded by:

$$\mathbb{E}[N_a^{\text{exploit}}(T)] \leq \frac{8 \log T}{\Delta_a} + C_a,$$

for some constant C_a that depends on initial pulls.

(3) Total Regret. The total regret is:

$$\text{Regret}(T) = \sum_{a \neq a^*} \Delta_a \cdot \left(\mathbb{E}[N_a^{\text{explore}}(T)] + \mathbb{E}[N_a^{\text{exploit}}(T)] \right),$$

yielding:

$$\text{Regret}(T) \leq \varepsilon T \cdot \Delta_{\max} + \sum_{a \neq a^*} \left(\frac{8 \log T}{\Delta_a} + C_a \right).$$

□

Extension to Decaying ε_t. If the exploration rate ε_t is annealed over time—e.g., $\varepsilon_t = \Theta(1/\sqrt{t})$—then the number of exploration rounds grows sublinearly in T, reducing the cumulative regret due to random exploration. In this setting, the overall regret becomes:

$$\text{Regret}(T) = O\left(\sum_{t=1}^{T} \varepsilon_t \cdot \Delta_{\max} \right) + O\left(\sum_{a \neq a^*} \frac{\log T}{\Delta_a} \right) = O\left(\sqrt{T} \cdot \Delta_{\max} + \sum_{a \neq a^*} \frac{\log T}{\Delta_a} \right).$$

For example, choosing $\varepsilon_t = \min\{1, c/\sqrt{t}\}$ ensures that exploration is significant early on but decays over time, enabling the agent to increasingly exploit the best empirical arm. This strategy achieves sublinear regret, improving over the linear regret of constant-ε policies while retaining simplicity and ease of implementation.

6.3.3 Regret of UCB.

UCB achieves strong performance guarantees. Assuming rewards are bounded in $[0, 1]$, UCB satisfies the following regret bound:

$$\text{Regret}(T) = O\left(\sum_{a \in \mathcal{A}: q(a) < q^*} \frac{\log T}{\Delta_a} \right),$$

where $q^* = \max_{a \in \mathcal{A}} q(a)$ is the optimal arm value, and $\Delta_a := q^* - q(a)$ is the suboptimality gap of arm a. This gap-dependent bound shows that UCB achieves near-optimal regret in the stochastic setting, with regret growing logarithmically in T and inversely with the difficulty Δ_a of distinguishing suboptimal arms from the best.

Theorem 6.3 (Regret of UCB)**.** *Consider a stochastic multi-armed bandit problem with $K = |\mathcal{A}|$ arms. Assume that the rewards for each arm $a \in \mathcal{A}$ are independent, drawn from distributions bounded in $[0, 1]$, and that the true mean rewards $q(a) = \mathbb{E}[R_t \mid A_t = a]$ are fixed but unknown.*

Let $q^* := \max_{a \in \mathcal{A}} q(a)$ denote the optimal arm value, and define the suboptimality gap for arm a as $\Delta_a := q^* - q(a)$. Let A_t denote the action selected at time t by the UCB algorithm:

$$A_t = \arg\max_{a \in \mathcal{A}} \left[\hat{q}_a(t) + \sqrt{\frac{2 \log t}{N_a(t)}} \right],$$

where $\hat{q}_a(t)$ is the empirical mean reward for arm a up to time t, and $N_a(t)$ is the number of times arm a has been selected.

Then the expected cumulative regret after T rounds satisfies:

$$\textit{Regret}(T) := \mathbb{E}\left[\sum_{t=1}^{T} (q^* - q(A_t)) \right] = O\left(\sum_{a \in \mathcal{A}: \Delta_a > 0} \frac{\log T}{\Delta_a} \right).$$

Proof Sketch. We analyze the regret of the UCB1 algorithm. Fix any suboptimal arm $a \in \mathcal{A}$ with suboptimality gap $\Delta_a := q^* - q(a) > 0$, where $q^* := \max_a q(a)$. Let $N_a(T)$ denote the number of times arm a is selected up to time T.

The regret incurred by pulling arm a is $\Delta_a \cdot N_a(T)$, so our goal is to bound $\mathbb{E}[N_a(T)]$.

Step 1: UCB selection condition. Suppose that at round t, arm a is chosen. Then its UCB must exceed that of the optimal arm a^*:

$$\hat{q}_a(t) + \sqrt{\frac{2\log t}{N_a(t)}} \geq \hat{q}_{a^*}(t) + \sqrt{\frac{2\log t}{N_{a^*}(t)}}.$$

Step 2: Concentration bounds. By Hoeffding's inequality, with high probability,

$$\hat{q}_a(t) \leq q(a) + \sqrt{\frac{2\log t}{N_a(t)}}, \qquad \hat{q}_{a^*}(t) \geq q^* - \sqrt{\frac{2\log t}{N_{a^*}(t)}}.$$

Combining with the selection condition gives:

$$q(a) + 2\sqrt{\frac{2\log t}{N_a(t)}} \geq q^* = q(a) + \Delta_a.$$

Rearranging:

$$2\sqrt{\frac{2\log t}{N_a(t)}} \geq \Delta_a \quad \Rightarrow \quad N_a(t) \leq \frac{8\log t}{\Delta_a^2}.$$

Step 3: Bounding expected pulls. We now integrate this high-probability bound into the expected value. After an initial number of explorations (say, one for each arm), the expected number of times a suboptimal arm a is pulled is bounded by:

$$\mathbb{E}[N_a(T)] \leq \frac{8\log T}{\Delta_a^2} + C,$$

for some constant C that accounts for the low-probability event in which the concentration bound fails.

Step 4: Regret bound. Summing over all suboptimal arms:

$$\text{Regret}(T) = \sum_{a:\Delta_a>0} \Delta_a \cdot \mathbb{E}[N_a(T)] = O\left(\sum_{a:\Delta_a>0} \frac{\log T}{\Delta_a}\right),$$

as claimed. □

6.3.4 Regret of Thompson Sampling.

Thompson Sampling also enjoys strong theoretical guarantees. For Bernoulli bandits, it achieves the regret bound:

$$\text{Regret}(T) = O\left(\sum_{a\in\mathcal{A}:q(a)<q^*} \frac{\log T}{\Delta_a}\right),$$

where $q^* = \max_{a\in\mathcal{A}} q(a)$ is the value of the best arm and $\Delta_a := q^* - q(a)$ is the suboptimality gap. This matches the performance of UCB up to constant factors. Recent analyses also establish minimax-optimal regret bounds under general reward distributions, making Thompson Sampling both theoretically and empirically effective.

Theorem 6.4 (Regret of Thompson Sampling for Bernoulli Bandits)**.** *Let $\mathcal{A} = \{1, \dots, K\}$ be a set of K arms, and suppose rewards are drawn independently from Bernoulli distributions with unknown means $q(a) \in [0, 1]$. Let $q^* = \max_{a \in \mathcal{A}} q(a)$ and define the suboptimality gap $\Delta_a := q^* - q(a)$ for each arm a.*

If Thompson Sampling is run with independent $\mathrm{Beta}(1, 1)$ *priors over the mean reward of each arm, then the expected regret after T rounds satisfies:*

$$\mathit{Regret}(T) = \mathbb{E}\left[\sum_{t=1}^{T}(q^* - q(A_t))\right] = O\left(\sum_{a \in \mathcal{A}:\Delta_a > 0} \frac{\log T}{\Delta_a}\right).$$

Proof. We present a sketch of the regret analysis for Thompson Sampling in the Bernoulli bandit setting, following the approach in [1].

Fix a suboptimal arm $a \in \mathcal{A}$ with suboptimality gap $\Delta_a := q^* - q(a) > 0$, where $q^* := \max_{a \in \mathcal{A}} q(a)$ is the mean reward of the optimal arm. Let $N_a(T)$ denote the number of times arm a is pulled by round T, and let $\theta_a(t) \sim \mathrm{Beta}(\alpha_a(t), \beta_a(t))$ be the posterior sample for arm a at time t, based on prior $\mathrm{Beta}(1, 1)$ and observed rewards.

The key idea is to bound the expected number of pulls of arm a by decomposing the event that Thompson Sampling selects arm a into three parts:

$$\mathbb{E}[N_a(T)] \leq \sum_{t=1}^{T} \mathbb{P}(A_t = a) \leq \sum_{t=1}^{T} \mathbb{P}(\theta_a(t) \geq \theta_{a^*}(t)).$$

We control the probability that $\theta_a(t)$ exceeds $\theta_{a^*}(t)$ by:

- Showing that $\theta_{a^*}(t)$ concentrates near q^* with high probability using concentration bounds for Beta distributions,
- Showing that $\theta_a(t)$ remains below $q^* - \Delta_a/2$ with high probability, provided arm a has been pulled sufficiently many times.

More precisely, with Hoeffding-type concentration for Beta posteriors, we can bound:

$$\mathbb{P}(\theta_a(t) > q(a) + \Delta_a/2) \leq \exp(-2N_a(t) \cdot \Delta_a^2),$$

and similarly for $\theta_{a^*}(t)$ being too far below q^*.

Integrating these bounds over time, we show that for each suboptimal arm a:

$$\mathbb{E}[N_a(T)] \leq O\left(\frac{\log T}{\Delta_a^2}\right).$$

Finally, the expected regret is:

$$\mathrm{Regret}(T) = \sum_{a \in \mathcal{A}:\Delta_a > 0} \Delta_a \cdot \mathbb{E}[N_a(T)] = O\left(\sum_{a:\Delta_a > 0} \frac{\log T}{\Delta_a}\right),$$

as claimed. □

6.3.5 Summary.

- Explore-then-Exploit is simple and analyzable, but not adaptive; regret is logarithmic with suboptimal constants.
- ε-greedy is easy to implement but may suffer high regret unless carefully tuned.

- UCB provides principled exploration and logarithmic regret in T.
- Thompson Sampling offers a Bayesian alternative with strong empirical and theoretical performance.
- These methods illustrate core ideas in balancing exploration with exploitation and lay the foundation for more general reinforcement learning algorithms.

6.4 Monte Carlo Prediction

Monte Carlo (MC) methods estimate value functions based on returns observed in complete episodes, without requiring knowledge of the transition model. These methods are especially suited for *episodic* tasks, where episodes eventually terminate.

6.4.1 Estimating v^π via Sampling

Let π be a fixed policy. The goal is to estimate the value function:

$$v^\pi(s) = \mathbb{E}_\pi \left[G_t \mid S_t = s \right],$$

where the return G_t is defined as the cumulative discounted reward from time t until the end of the episode:

$$G_t := \sum_{k=t}^{T-1} \gamma^{k-t} R_{k+1}.$$

Here, T is the (random) time when the episode terminates.

Monte Carlo methods estimate $v^\pi(s)$ via empirical averages over sampled episodes:

$$\hat{v}_\pi(s) = \frac{1}{N(s)} \sum_{i=1}^{N(s)} G_t^{(i)},$$

where $G_t^{(i)}$ is the return observed from the i-th visit to state s, and $N(s)$ is the total number of such visits.

Incremental Monte Carlo Update. Rather than storing all returns, we can update $\hat{v}_\pi(s)$ incrementally after each visit to state s. Let $N(s)$ denote the number of times s has been visited (including the current one), and let G_t be the observed return from that visit. Then the update is:

$$\begin{aligned} N(s) &\leftarrow N(s) + 1, \\ \hat{v}_\pi(s) &\leftarrow \hat{v}_\pi(s) + \frac{1}{N(s)} \left(G_t - \hat{v}_\pi(s) \right). \end{aligned}$$

This rule incrementally computes the sample average and is equivalent to the standard Monte Carlo estimate:

$$\hat{v}_\pi(s) = \frac{1}{N(s)} \sum_{i=1}^{N(s)} G_t^{(i)}.$$

6.4.2 State Visits and Return Estimation

We define a *visit* to state s as any time step t where $S_t = s$. Since a state may be visited multiple times within an episode, different strategies exist for estimating $v^\pi(s)$.

First-Visit Monte Carlo. Only the first occurrence of each state in an episode is used to compute its return:

$$\hat{v}_\pi^{\mathrm{FV}}(s) = \frac{1}{N_{\mathrm{FV}}(s)} \sum_{i=1}^{N_{\mathrm{FV}}(s)} G_{t_i},$$

where t_i is the first time s appears in the i-th episode. This estimator uses independent samples across episodes, which is desirable for theoretical analysis.

Every-Visit Monte Carlo. All visits to a state are used:

$$\hat{v}_\pi^{\mathrm{EV}}(s) = \frac{1}{N_{\mathrm{EV}}(s)} \sum_{j=1}^{N_{\mathrm{EV}}(s)} G_{t_j},$$

where t_j indexes all time steps in all episodes for which $S_{t_j} = s$. This approach uses more data, improving efficiency, but the samples are no longer independent.

First-visit MC prediction, for estimating $V \approx v_\pi$

Input: a policy π to be evaluated
Initialize:
    $V(s) \in \mathbb{R}$, arbitrarily, for all $s \in \mathcal{S}$
    $Returns(s) \leftarrow$ an empty list, for all $s \in \mathcal{S}$

Loop forever (for each episode):
    Generate an episode following π: $S_0, A_0, R_1, S_1, A_1, R_2, \ldots, S_{T-1}, A_{T-1}, R_T$
    $G \leftarrow 0$
    Loop for each step of episode, $t = T-1, T-2, \ldots, 0$:
        $G \leftarrow \gamma G + R_{t+1}$
        Unless S_t appears in $S_0, S_1, \ldots, S_{t-1}$:
            Append G to $Returns(S_t)$
            $V(S_t) \leftarrow$ average$(Returns(S_t))$

Figure 6.1: Pseudocode for the First-Visit Monte Carlo prediction algorithm. The return is computed in reverse order at the end of each episode, and each state's value estimate is updated only on its first visit within the episode.

6.4.3 Illustrative Example

Consider the following trajectory generated by a random policy in an MDP with discount factor γ:

$$\begin{aligned}
t = 0, &\quad s_1, a_r \to s_2, &\quad r_1 = 0 \\
t = 1, &\quad s_2, a_r \to s_3, &\quad r_2 = 0 \\
t = 2, &\quad s_3, a_l \to s_2, &\quad r_3 = 0 \\
t = 3, &\quad s_2, a_r \to s_3, &\quad r_4 = 0 \\
t = 4, &\quad s_3, a_r \to s_g, &\quad r_5 = 10
\end{aligned}$$

The state and reward sequence is:

$$(s_1, s_2, s_3, s_2, s_3, s_g), \quad (0, 0, 0, 0, 10)$$

Return Computation. We compute the returns G_t from each timestep:

$$\begin{aligned} G_4 &= 10, \\ G_3 &= 0 + \gamma \cdot G_4 = \gamma \cdot 10, \\ G_2 &= 0 + \gamma \cdot G_3 = \gamma^2 \cdot 10, \\ G_1 &= 0 + \gamma \cdot G_2 = \gamma^3 \cdot 10, \\ G_0 &= 0 + \gamma \cdot G_1 == \gamma^4 \cdot 10, \\ &\cdot \end{aligned}$$

First-Visit MC Estimate.

- First visit to s_1 is at $t = 0$: $\hat{v}_\pi^{\text{FV}}(s_1) = G_0 = \gamma^4 \cdot 10$
- First visit to s_2 is at $t = 1$: $\hat{v}_\pi^{\text{FV}}(s_2) = G_1 = \gamma^3(2 + \gamma^2 \cdot 10)$
- First visit to s_3 is at $t = 2$: $\hat{v}_\pi^{\text{FV}}(s_3) = G_2 = \gamma(2 + \gamma^2 \cdot 10)$

Every-Visit MC Estimate.

- s_1 is visited once: $\hat{v}_\pi^{\text{EV}}(s_1) = G_0$
- s_2 is visited twice at $t = 1$ and $t = 3$: $\hat{v}_\pi^{\text{EV}}(s_2) = \frac{1}{2}(G_1 + G_3)$
- s_3 is visited twice at $t = 2$ and $t = 4$: $\hat{v}_\pi^{\text{EV}}(s_3) = \frac{1}{2}(G_2 + G_4)$

This illustrates how first-visit MC uses only the first occurrence per episode, while every-visit MC aggregates across all visits, leading to more data but possibly higher bias due to sample dependence.

6.4.4 Comparison and Discussion

Both first-visit and every-visit Monte Carlo methods are unbiased in expectation and converge to the true value function v^π as the number of visits $N(s) \to \infty$. However, they differ in statistical properties that affect convergence speed and variance:

- **First-Visit MC:**
  - Uses only the first occurrence of each state per episode,
  - Produces independent return samples (one per episode),
  - Higher variance due to fewer samples.
- **Every-Visit MC:**
  - Uses all occurrences of a state within an episode,
  - Samples are not independent, and may introduce mild bias,
  - Typically achieves lower variance in practice.

This tradeoff is illustrated in Figure 6.2, which shows that every-visit estimators typically reduce the root mean squared error more quickly, especially in early stages of learning. However, the independence of first-visit samples makes them easier to analyze theoretically and useful for establishing convergence guarantees.

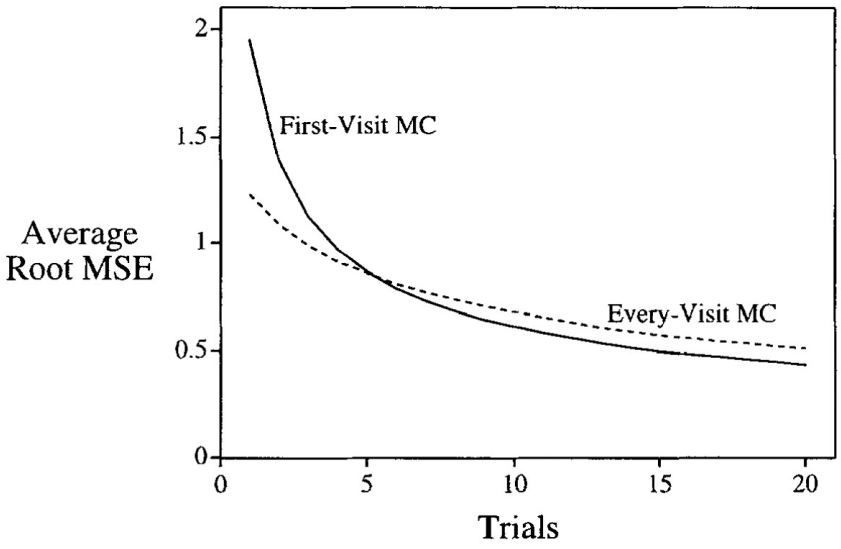

Figure 6.2: Empirical comparison of first-visit and every-visit Monte Carlo estimators. Every-visit tends to converge faster with lower mean squared error, though at the cost of dependent samples. Adapted from Singh & Sutton (1996).

Estimator	**Bias**	**Variance**	**Sample Independence**
First-Visit	Unbiased	Higher	Yes
Every-Visit	Unbiased	Lower	No

Table 6.1: Comparison of First-Visit and Every-Visit Monte Carlo estimators.

6.4.5 Advantages and Limitations

- **Advantages:**
  - Simple to implement.
  - Model-free: does not require knowledge of transitions or rewards.
  - Converges under the Law of Large Numbers.
- **Limitations:**
  - Requires episodes to terminate.
  - High variance in long episodes.
  - Inefficient for continuing tasks.

This avoids storing the full list of returns and only requires maintaining:
- the current value estimate $V(s)$,
- the count of visits $N(s)$.

Advantages.
- Constant memory and runtime per update.
- Particularly useful in long-running or online settings.

- Easily generalizes to $Q(s, a)$ updates in control algorithms.

This incremental scheme is widely used in practical implementations of Monte Carlo prediction and control, and forms the foundation for further extensions like TD learning and Q-learning.

6.5 Monte Carlo Control

So far we have used Monte Carlo methods to evaluate a fixed policy. We now turn to the problem of finding an optimal policy through experience—using only samples collected from interaction with the environment. This setting, where we do not assume access to a model, is referred to as *model-free control*.

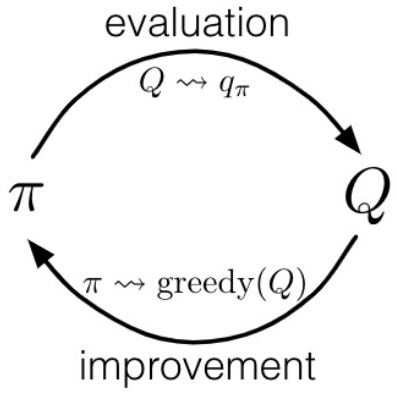

Figure 6.3: Generalized Policy Iteration (GPI): Monte Carlo control alternates between evaluating a policy and improving it based on current estimates.

Generalized Policy Iteration (GPI). Monte Carlo control implements the GPI framework: given an estimate $\hat{q}_{\pi_k} \approx q_{\pi_k}$, we improve the policy via greedy selection:

$$\pi_{k+1}(s) = \arg\max_a \hat{q}_{\pi_k}(s, a).$$

$$\pi_0 \xrightarrow{\text{E}} q_{\pi_0} \xrightarrow{\text{I}} \pi_1 \xrightarrow{\text{E}} q_{\pi_1} \xrightarrow{\text{I}} \pi_2 \xrightarrow{\text{E}} \cdots \xrightarrow{\text{I}} \pi_* \xrightarrow{\text{E}} q_*,$$

Figure 6.4: Monte Carlo control viewed as an iterative GPI process: evaluation (E) and improvement (I) steps alternate until convergence.

6.5.1 Monte Carlo Control with Exploring Starts

The MC-ES algorithm guarantees convergence to the optimal policy under the assumption that every state-action pair (s, a) is visited infinitely often. This is ensured by selecting episodes with *exploring starts*—that is, starting from any (s, a) pair with positive probability.

While theoretically sound, exploring starts are impractical in many real-world environments, where one cannot freely choose arbitrary start states or actions. This motivates an alternative based on *on-policy* exploration.

Monte Carlo ES (Exploring Starts), for estimating $\pi \approx \pi_*$

Initialize:
  $\pi(s) \in \mathcal{A}(s)$ (arbitrarily), for all $s \in \mathcal{S}$
  $Q(s,a) \in \mathbb{R}$ (arbitrarily), for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$
  $Returns(s,a) \leftarrow$ empty list, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$

Loop forever (for each episode):
  Choose $S_0 \in \mathcal{S}$, $A_0 \in \mathcal{A}(S_0)$ randomly such that all pairs have probability > 0
  Generate an episode from S_0, A_0, following π: $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$
  $G \leftarrow 0$
  Loop for each step of episode, $t = T-1, T-2, \ldots, 0$:
    $G \leftarrow \gamma G + R_{t+1}$
    Unless the pair S_t, A_t appears in $S_0, A_0, S_1, A_1 \ldots, S_{t-1}, A_{t-1}$:
      Append G to $Returns(S_t, A_t)$
      $Q(S_t, A_t) \leftarrow$ average$(Returns(S_t, A_t))$
      $\pi(S_t) \leftarrow \arg\max_a Q(S_t, a)$

Figure 6.5: Monte Carlo control with exploring starts (ES): guarantees convergence under the assumption that all state-action pairs are visited with nonzero probability.

6.5.2 Exploration via Soft Policies

To ensure convergence without access to arbitrary start states, we instead enforce sufficient exploration through *soft policies*—those that assign nonzero probability to all actions in every state. A canonical example is the ε-greedy policy.

Definition 6.1 (GLIE)**.** *A sequence of policies (π_t) is said to satisfy the* GLIE *(Greedy in the Limit with Infinite Exploration) conditions if:*

- *Every action is taken infinitely often in every state:*

$$\lim_{t\to\infty} N_t(s,a) = \infty, \quad \forall (s,a),$$

- *The policy becomes greedy in the limit:*

$$\lim_{t\to\infty} \pi_t(a \mid s) = \begin{cases} 1 & \text{if } a \in \arg\max_{a'} Q_t(s,a'), \\ 0 & \text{otherwise.} \end{cases}$$

This condition is often satisfied by using an ε-greedy policy with decaying ε_t, e.g., $\varepsilon_t = 1/\sqrt{t}$.

6.5.3 On-Policy MC Control with ε-Soft Policies

This algorithm improves upon MC-ES by using an ε-soft policy. At each episode:

1. Generate an episode using the current ε-soft policy π_t.
2. For each (s,a) in the episode, compute the return G_t and update $Q(s,a)$ using first-visit Monte Carlo.
3. Update π_t to be ε_t-greedy with respect to Q.

On-policy first-visit MC control (for ε-soft policies), estimates $\pi \approx \pi_*$

Algorithm parameter: small $\varepsilon > 0$
Initialize:
  $\pi \leftarrow$ an arbitrary ε-soft policy
  $Q(s,a) \in \mathbb{R}$ (arbitrarily), for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$
  $Returns(s,a) \leftarrow$ empty list, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$

Repeat forever (for each episode):
  Generate an episode following π: $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$
  $G \leftarrow 0$
  Loop for each step of episode, $t = T-1, T-2, \ldots, 0$:
    $G \leftarrow \gamma G + R_{t+1}$
    Unless the pair S_t, A_t appears in $S_0, A_0, S_1, A_1 \ldots, S_{t-1}, A_{t-1}$:
      Append G to $Returns(S_t, A_t)$
      $Q(S_t, A_t) \leftarrow \text{average}(Returns(S_t, A_t))$
      $A^* \leftarrow \arg\max_a Q(S_t, a)$ (with ties broken arbitrarily)
      For all $a \in \mathcal{A}(S_t)$:

$$\pi(a|S_t) \leftarrow \begin{cases} 1 - \varepsilon + \varepsilon/|\mathcal{A}(S_t)| & \text{if } a = A^* \\ \varepsilon/|\mathcal{A}(S_t)| & \text{if } a \neq A^* \end{cases}$$

Figure 6.6: On-policy first-visit Monte Carlo control using ε-soft policies. Ensures sufficient exploration through stochasticity while converging to a greedy policy in the limit.

6.5.4 Convergence Guarantees

Theorem 6.5 (Convergence of Monte Carlo Control)**.** *Suppose each episode is generated by an ε-greedy policy with $\varepsilon_t \to 0$ and the resulting sequence of policies (π_t) satisfies GLIE. Then:*

$$\lim_{t\to\infty} Q_t(s,a) = q^*(s,a), \quad \textit{and} \quad \lim_{t\to\infty} \pi_t = \pi^*,$$

with probability 1, for all (s,a).

Remarks.

- Monte Carlo control requires episodic tasks with complete returns from each episode.
- GLIE ensures sufficient exploration and eventual exploitation.
- In practice, convergence can be slow; temporal-difference methods (e.g., SARSA, Q-learning) often provide faster alternatives.

Concentration of Beta Posterior for Bernoulli Rewards

Lemma .1 (Concentration of Beta Posterior for Bernoulli Rewards)**.** *Let $X_1, \ldots, X_n \sim \text{Bernoulli}(q)$ be i.i.d. observations with mean $q \in [0,1]$, and let $\hat{q}_n := \frac{1}{n}\sum_{i=1}^n X_i$ be the empirical mean. Suppose the prior over q is $\text{Beta}(\alpha, \beta)$, and let $\theta \sim \text{Beta}(\alpha + \sum X_i, \beta + n - \sum X_i)$ be a sample from the posterior. Then for any $\epsilon > 0$,*

$$\mathbb{P}\left(\theta \geq \hat{q}_n + \epsilon\right) \leq \exp(-2n\epsilon^2), \quad \textit{and} \quad \mathbb{P}\left(\theta \leq \hat{q}_n - \epsilon\right) \leq \exp(-2n\epsilon^2).$$

Proof. Let $S_n := \sum_{i=1}^n X_i$. The posterior distribution after observing $X_1, \ldots, X_n$ is:

$$\theta \sim \text{Beta}(\alpha + S_n, \beta + n - S_n).$$

This distribution has mean

$$\mathbb{E}[\theta] = \frac{\alpha + S_n}{\alpha + \beta + n} = \frac{\alpha + n\hat{q}_n}{\alpha + \beta + n},$$

and variance

$$\text{Var}(\theta) = \frac{(\alpha + S_n)(\beta + n - S_n)}{(\alpha + \beta + n)^2(\alpha + \beta + n + 1)}.$$

To bound the probability that θ deviates from the empirical mean $\hat{q}_n$, we use the fact that the posterior is a smooth distribution centered near $\hat{q}_n$, and concentrate our attention on deviations due to randomness in the sample.

By Hoeffding's inequality, for the empirical mean $\hat{q}_n = \frac{S_n}{n}$, we have:

$$\mathbb{P}\left(|\hat{q}_n - q| > \epsilon\right) \leq 2\exp(-2n\epsilon^2).$$

Now, observe that:

$$|\theta - q| \leq |\theta - \hat{q}_n| + |\hat{q}_n - q|.$$

By the triangle inequality, if both $|\theta - \hat{q}_n| \leq \epsilon/2$ and $|\hat{q}_n - q| \leq \epsilon/2$, then $|\theta - q| \leq \epsilon$. Hence, if $|\theta - q| > \epsilon$, then at least one of these two deviations must exceed $\epsilon/2$.

So we can write:

$$\mathbb{P}(|\theta - \hat{q}_n| > \epsilon) \leq \mathbb{P}(|\theta - q| > \epsilon/2) + \mathbb{P}(|\hat{q}_n - q| > \epsilon/2).$$

Since θ is a random variable with mean close to $\hat{q}_n$ and concentrates as n grows, we can treat both terms with similar concentration behavior. Using Hoeffding's inequality again on the second term:

$$\mathbb{P}(|\hat{q}_n - q| > \epsilon/2) \leq 2\exp(-n\epsilon^2/2).$$

And using the fact that the Beta posterior is sharply concentrated around $\hat{q}_n$, we obtain (see e.g., [Agrawal and Goyal 2012]):

$$\mathbb{P}(|\theta - \hat{q}_n| > \epsilon) \leq 2\exp(-2n\epsilon^2),$$

as desired. □

References for Lecture

[1] Shipra Agrawal and Navin Goyal. "Analysis of Thompson Sampling for the Multi-Armed Bandit Problem". In: *Proceedings of the 25th Annual Conference on Learning Theory (COLT)*. 2012, pp. 39.1–39.26.