Lecture 7: Bayesian & Information-Theoretic Exploration

Thompson Sampling, Information-Directed Sampling, and Posterior Sampling for RL

1 Introduction

1.1 Where we are

Last lecture, we covered:

- Why exploration matters (sigmoid scaling curves, sparse reward).
- Entropy regularization and its role in convergence theory.
- Practical exploration methods: parameter space noise, intrinsic motivation, count-based bonuses.
- Multi-armed bandits, regret, ϵ-greedy, and UCB.

Today we cover two exploration paradigms: **Thompson sampling** and **information-theoretic methods**. We develop these first in the bandit setting, prove key regret bounds, and then show how they scale to full RL.

Roadmap:

1. **Thompson sampling** — the Bayesian approach to exploration (Section 2).
2. **Scalable posterior approximations** — Bootstrap DQN, EpiNets, and hypermodels (Sections 3 and 4).
3. **Posterior sampling for RL (PSRL)** — extending Thompson sampling from bandits to full MDPs (Section 5).
4. **Information-theoretic exploration** — information-directed sampling and connections to Bayesian experimental design (Section 6).

Logistics:

1. Homework 2+3 are posted, let TAs know if there are issues/questions.
2. Project proposals due today! You're not locked into the idea, so just post it even if you're unsure.
3. Don't forget to post feedback on others' projects too due in a week!
4. Got lots of last minute requests to discuss projects, so we will end class a bit early today so people can discuss with myself or TAs here today.

2 Thompson Sampling

Recall the multi-armed bandit.

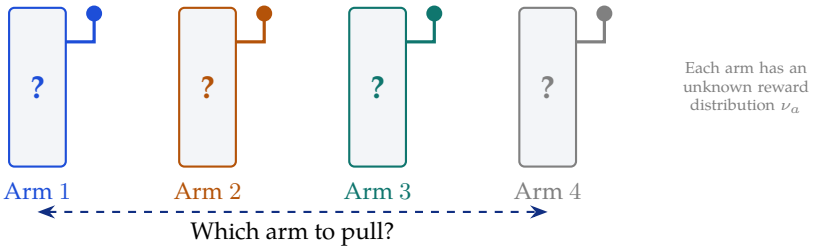

Figure 1: The K-armed bandit problem ($K=4$). Each arm has an unknown reward distribution. The agent must decide which arm to pull at each round, balancing exploration (learning about uncertain arms) and exploitation (pulling the arm believed to be best).

Thompson sampling (Thompson, 1933) is another mechanism for navigating the exploration–exploitation tradeoff. The idea: sample each action with probability proportional to the probability that it is optimal. Estimating that probability is often easier said than done (see, e.g., LLMs), but we'll get back to that later.

Consider a general online decision problem with unknown parameter θ^* drawn from a prior $p(\theta)$. At each time step t, the agent selects an action $a_t \in \mathcal{A}$, observes a reward r_t drawn from a distribution that depends on a_t and θ^*, and updates a posterior over θ^*. Let $\mathcal{H}_t = (a_1, r_1, \ldots, a_{t-1}, r_{t-1})$ denote the history. The posterior after $t-1$ observations is:

$$p(\theta \mid \mathcal{H}_t) \propto p(\theta) \prod_{\tau=1}^{t-1} p(r_\tau \mid a_\tau, \theta). \tag{1}$$

Thompson sampling is: sample a parameter from the posterior, then act as if that sample were the truth.

Algorithm 3 Thompson Sampling (General)

Input: Prior $p(\theta)$, action set $\mathcal{A}$.
for $t = 1, 2, \ldots, T$ **do**

Sample $\tilde{\theta}_t \sim p(\theta \mid \mathcal{H}_t)$. ▷ Draw from current posterior
Select $a_t = \arg\max_{a \in \mathcal{A}} \mathbb{E}[r \mid a, \tilde{\theta}_t]$. ▷ Act optimally under sampled model
Observe reward r_t and update posterior.

2.1 Concrete example: Bernoulli bandits

Consider a K-armed bandit where each arm a gives reward $r \sim \text{Bernoulli}(\mu_a)$. A Bernoulli random variable takes value 1 with probability μ_a and 0 with probability $1-\mu_a$, like a biased coin flip.

We place a Beta prior on each arm:

$$\mu_a \sim \text{Beta}(\alpha_a, \beta_a). \tag{2}$$

The Beta distribution is a distribution on $[0, 1]$ parameterized by $\alpha, \beta > 0$, with mean $\alpha/(\alpha + \beta)$. It is the conjugate prior for the Bernoulli likelihood, meaning the posterior stays in the Beta family after each observation.

Initially, $\alpha_a = \beta_a = 1$ (uniform prior). After observing S_a successes and F_a failures on arm a, the posterior is:

$$\mu_a \mid \mathcal{D} \sim \text{Beta}(\alpha_a + S_a,\ \beta_a + F_a). \tag{3}$$

Algorithm 4 Thompson Sampling for Bernoulli Bandits

Initialize $\alpha_a = 1, \beta_a = 1$ for all arms a.
for $t = 1, 2, \ldots, T$ **do**

for each arm a **do**: sample $\tilde{\mu}_a \sim \text{Beta}(\alpha_a, \beta_a)$.
Play arm $a_t = \arg\max_a \tilde{\mu}_a$.
Observe reward $r_t \in \{0, 1\}$.
Update: $\alpha_{a_t} \leftarrow \alpha_{a_t} + r_t, \quad \beta_{a_t} \leftarrow \beta_{a_t} + (1 - r_t)$.

Figure 2 shows how the Beta posteriors evolve. Early on, wide posteriors overlap, so Thompson sampling tries all arms. As data accumulates, the posteriors concentrate and TS exploits the best arm.

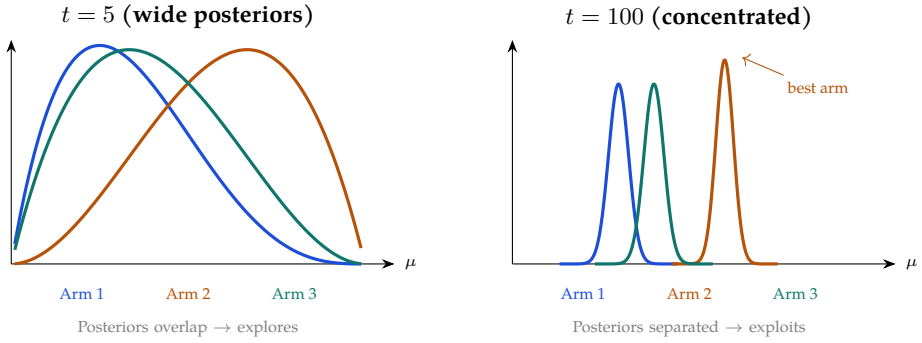

Figure 2: Beta posteriors for a 3-armed Bernoulli bandit. **Left:** After 5 rounds, posteriors are wide and overlapping. Different arms are frequently sampled as the best, driving exploration. **Right:** After 100 rounds, posteriors have concentrated. Arm 2 (orange) is almost always sampled as the best.

We won't get into it, but if you're interested, you can look to Agrawal and Goyal (2012); Kaufmann et al. (2012) to see that Bernoulli bandits with Thompson sampling and Beta priors achieve bounded regret. Chapelle and Li (2011) found that Thompson sampling often outperforms UCB in practice, despite comparable theoretical guarantees. Thompson sampling adapts to the problem structure: it explores more when the posterior is diffuse and less when it is concentrated, while UCB's exploration bonus can be overly conservative.

3 Thompson Sampling for RL: Bootstrap DQN

Thompson sampling requires maintaining a posterior over the unknown quantities. In bandits, these are reward means (low-dimensional). In RL, the unknown is the *Q-function* (enormous). How can we scale Thompson sampling to deep RL (or at least some Thompson sampling-like approach)?

Osband et al. (2016) propose the following approximation. Instead of maintaining a full posterior over Q-functions, represent it as an **ensemble** of k Q-networks:

$$p(Q \mid \mathcal{D}) \approx \frac{1}{k} \sum_{i=1}^{k} \delta(Q = Q_{\theta_i}). \tag{4}$$

Each Q-network has the same architecture but different random initialization. The algorithm is:

Algorithm 5 Bootstrap DQN (Osband et al., 2016)

Initialize k Q-networks $Q_{\theta_1}, \ldots, Q_{\theta_k}$ with random weights.
for each episode **do**

Sample $i \sim \text{Uniform}(\{1, \ldots, k\})$. ▷ Thompson sampling-like step
Collect an episode acting greedily w.r.t. Q_{θ_i}: $a_t = \arg\max_a Q_{\theta_i}(s_t, a)$.
Store transitions in replay buffer $\mathcal{D}$.
Update all k Q-networks using transitions from $\mathcal{D}$.

The ensemble captures **epistemic uncertainty**: uncertainty due to limited data, which can be reduced by collecting more data. This is distinct from **aleatoric uncertainty**: inherent stochasticity in the environment, which cannot be reduced.

- **Early training**: The Q-networks disagree substantially (they had different random initializations and have seen limited data). Sampling a Q-network and acting greedily produces diverse, temporally coherent exploration strategies.
- **Late training**: All Q-networks converge to similar estimates (they have all seen enough data). Sampling any Q-network produces similar behavior, and exploration decreases to near zero.

Discussion

Compare and contrast this with other exploration approaches we've seen so far in Q learning. When do you think this might work well? What about not so well?

4 A common theme in uncertainty sampling and a broader note on ensemble methods

The Bootstrap DQN approach from the previous section is one instance of a broader idea that you can use ensembling to estimate uncertainty and posteriors. Another common approach is to use test-time dropout. Gal and Ghahramani (2016) show that dropout training can be interpreted

as approximate Bayesian inference: at test time, running multiple forward passes with dropout active produces samples from an approximate posterior over network outputs. This **MC Dropout** technique can be used for Thompson sampling-style exploration by sampling a dropout mask and acting greedily with respect to the resulting Q-values. However, Osband et al. (2016) found that MC Dropout produces "spiky" posterior samples that do not resemble plausible Q-functions, making it less effective than ensembles for deep exploration in practice.

Epistemic neural networks (EpiNets). Osband et al. (2023) propose a lightweight alternative to ensembles. An **epinet** is a small auxiliary network that takes as additional input a random "epistemic index" $z \sim P_Z$ and outputs a perturbation to the base network's predictions:

$$Q(s, a; z) = f_\theta(s, a) + \sigma \cdot e_\phi(s, a, z), \tag{5}$$

where f_θ is a conventional (pretrained) network, e_ϕ is the epinet, and σ controls the scale. Different draws of z produce different Q-function "samples," enabling Thompson sampling. EpiNets allegedly match or outperform large ensembles (hundreds of members) while adding only small amounts of computational overhead.

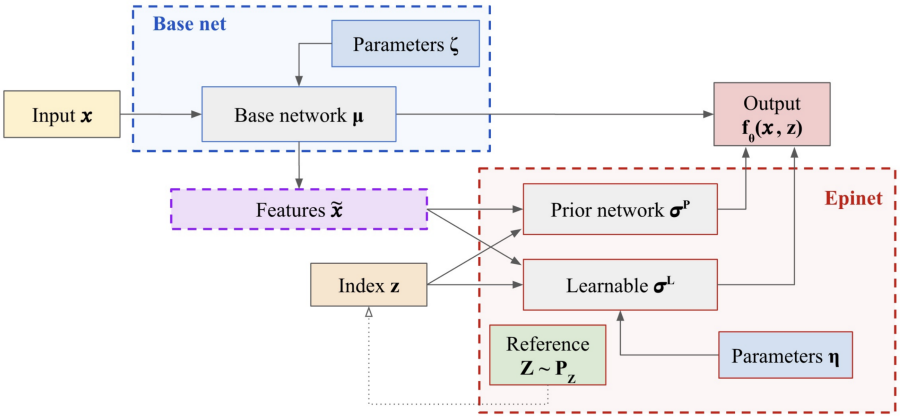

Figure 3: The epinet architecture (Osband et al., 2023). A conventional base network f_θ is augmented with a small epinet e_ϕ that takes an additional random epistemic index z as input. Different draws of z produce different output perturbations, approximating posterior samples without maintaining a full ensemble.

5 Posterior Sampling for Reinforcement Learning (PSRL)

Thompson sampling was originally defined for bandits. **Posterior Sampling for Reinforcement Learning** (PSRL), proposed by Strens (2000); Osband et al. (2013), extends the idea to episodic MDPs.

5.1 Setup: Bayesian RL

Consider an episodic MDP $M = (\mathcal{S}, \mathcal{A}, P, R, H, \rho)$ with state space $\mathcal{S}$, action space $\mathcal{A}$, transition kernel P, reward function R, horizon H, and initial state distribution ρ. The agent interacts with the MDP for K episodes.

In the Bayesian setting, the transition function $\mathcal{T}_\theta$ is unknown, parameterized by some latent $\theta \in \Theta$. The agent's prior uncertainty over θ is captured by a distribution $p(\theta)$, which is updated to a posterior $p(\theta \mid \mathcal{H}_k)$ after each episode using Bayes' rule, where $\mathcal{H}_k$ is the history of all transitions observed so far.

We can formalize this as a **Bayes-Adaptive MDP** (BAMDP) (Duff, 2002). The BAMDP is defined over a **hyperstate** space $\mathcal{X} = \mathcal{S} \times \Delta(\Theta)$. A hyperstate $x = \langle s, p \rangle$ pairs the agent's physical state $s \in \mathcal{S}$ with its **epistemic state** $p \in \Delta(\Theta)$, a distribution over Θ encoding everything the agent knows about the environment from previously observed data (Lu et al., 2023). The BAMDP transition function updates both components: s transitions according to the true (unknown) MDP, and p is updated via Bayes' rule given the observed transition (s, a, s'). The BAMDP reward function is the same as the original MDP's: $\bar{\mathcal{R}}(\langle s, p \rangle, a) = \mathcal{R}(s, a)$. Though, as a note, you can also have a BAMDP with unknown reward also parameterized by θ, maintaining a posterior over this as well.

A BAMDP policy $\pi_h : \mathcal{X} \to \mathcal{A}$ selects actions based on both the physical state and the epistemic state. The Bayes-optimal policy would minimize expected cumulative regret, but computing it requires planning over the full hyperstate space, which is intractable for all but the smallest problems (the epistemic state is continuous and grows in complexity with each observation). PSRL sidesteps this by sampling a single MDP from the posterior and planning in that MDP instead.

5.2 The PSRL algorithm

Algorithm 6 Posterior Sampling for RL (PSRL) (Osband et al., 2013)

Input: Prior $p(M)$ over MDPs.
for episode $k = 1, 2, \ldots, K$ **do**

Sample $\tilde{M}_k \sim p(M \mid \mathcal{H}_k)$. ▷ Sample an MDP from the posterior
Compute optimal policy $\tilde{\pi}_k = \pi^*(\tilde{M}_k)$. ▷ Solve the sampled MDP
Execute $\tilde{\pi}_k$ for one episode, collecting trajectory $(s_1^k, a_1^k, r_1^k, \ldots, s_H^k, a_H^k, r_H^k)$.
Update posterior: $p(M \mid \mathcal{H}_{k+1})$.

The structure mirrors Thompson sampling: sample from the posterior, act optimally under the sample. The difference is that in RL, "acting optimally" means solving an entire MDP (via dynamic programming) rather than picking a single arm. Figure 4 illustrates this cycle.

Discussion

Comparing this with ϵ-greedy or even bandit-based Thompson sampling or Bootstrap Q Learning? What do you notice about the exploration here? Discuss the differences with your neighbors. Also discuss why might it be hard to operationalize? Think LLMs.

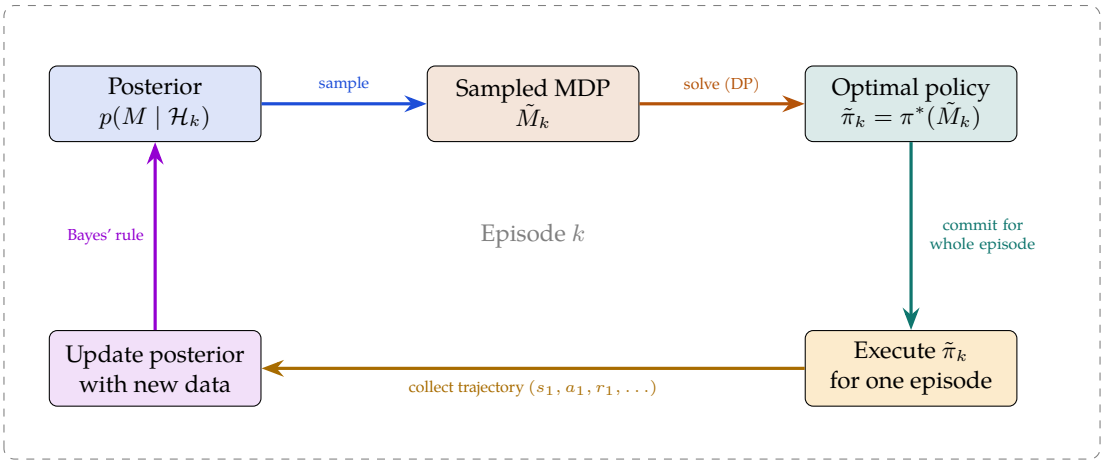

Figure 4: The PSRL cycle. Each episode: (1) sample an MDP from the posterior, (2) solve it via dynamic programming, (3) commit to the optimal policy for the entire episode, (4) collect data and update the posterior. The key difference from Thompson sampling in bandits is step 2: solving a full MDP instead of picking a single arm.

5.3 LLMs as posterior samplers

Arumugam and Griffiths (2025) show that LLMs can directly implement PSRL in natural-language environments without modifying the core algorithm. The architecture uses:

1. One LLM prompt/memory to maintain a *textual posterior* over environment dynamics.
2. Another to sample plausible hypotheses (the posterior sampling step).
3. A third LLM prompt to act optimally under that sampled hypothesis (the planning step).

This lets LLM agents inherit the statistically efficient exploration guarantees of PSRL while leveraging the generalization capabilities of LLMs. Empirically, LLM-based PSRL substantially outperforms naive LLM agents on tasks requiring prudent exploration (multi-armed bandits, Wordle, and other sequential decision-making tasks). This also provides one of the first empirical validations of PSRL beyond toy tabular settings.

Implications for Frontiers

Why do I think this is exciting? It gives us a pathway to thinking about compressed representations of posteriors. And it combines with a long line of research combining Bayesian methods with LLMs. But, what do you make of this? Do you think LLMs can verbalize beliefs over MDPs? What about complex MDPs?

6 Information-Theoretic Exploration

Having covered posterior sampling methods — from Thompson sampling for bandits, through Bootstrap DQN and its practical variants, to PSRL for full MDPs — we now turn to a complementary

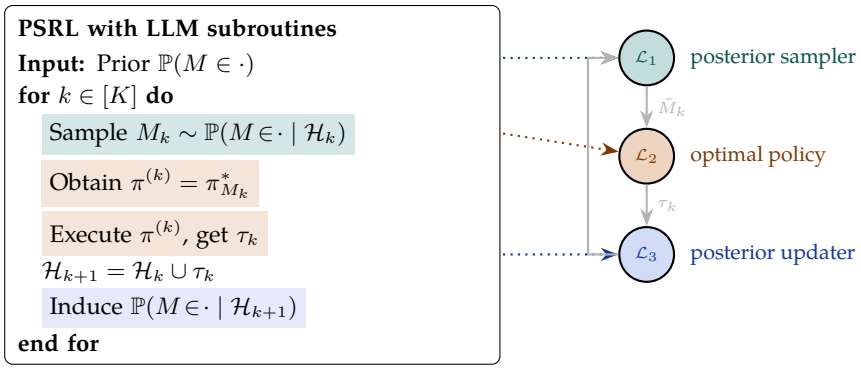

Figure 5: LLM-based PSRL (Arumugam and Griffiths, 2025), after their Figure 2. Each step of the PSRL algorithm is outsourced to an LLM: **posterior sampling** (draw a plausible MDP from a textual belief state), **optimal policy** (act under the sampled hypothesis), and **posterior updating** (revise beliefs after observing a trajectory). More capable LLMs (e.g., o1-mini vs. GPT-4o) maintain more accurate textual posteriors, yielding sublinear rather than linear regret.

paradigm: **information-theoretic exploration**. Rather than sampling from beliefs, these methods directly ask: *what action gives us the most useful information?*

Discussion

Turn to your neighbor and think about when Thompson sampling might be suboptimal. What happens when actions provide different amounts of information?

6.1 Information-Directed Sampling (IDS)

6.1.1 Motivation: when Thompson sampling is suboptimal

Thompson sampling is near-optimal for many standard problems, but it can be suboptimal when actions provide *different amounts of information*. Consider the following example:

Example 1 (Sparse linear bandits). Suppose $\theta^* \in \{0,1\}^d$ and the action set includes:

- d "informative" actions $e_1, \ldots, e_d$ (standard basis vectors) that each reveal one component of θ^*.
- One "rewarding" action $a^* = (1,1,\ldots,1)/\sqrt{d}$ that has high reward when $\|\theta^*\|$ is large, but reveals little about individual components.

Thompson sampling may waste time playing a^* (which it often samples as optimal) when the informative actions e_i would be more useful.

6.1.2 The IDS algorithm

Russo and Van Roy (2014) propose **information-directed sampling**, which explicitly trades off regret and information. "The act of sacrificing immediate reward for delayed information is sometimes referred to as deep exploration" (Lu et al., 2023; Osband et al., 2019). IDS directly minimizes the *information ratio* at each time step.

We need two ingredients. Let $\Delta_t(a)$ denote the expected instantaneous regret of action a at time t:

$$\Delta_t(a) := \mathbb{E}\big[\mu^*(\theta^*) - r_t \mid \mathcal{H}_t,\, a_t = a\big], \tag{6}$$

and let $g_t(a)$ denote the expected information gain about the optimal action A^* from playing a:

$$g_t(a) := \mathbb{E}\big[H(\alpha_t) - H(\alpha_{t+1}) \mid \mathcal{H}_t,\, a_t = a\big], \tag{7}$$

where α_t is the posterior distribution over A^* and $H(\cdot)$ is Shannon entropy. For a sampling distribution $\pi \in \mathcal{D}(\mathcal{A})$, we overload the notation: $\Delta_t(\pi) = \sum_a \pi(a)\Delta_t(a)$ and $g_t(\pi) = \sum_a \pi(a) g_t(a)$.

Definition 1 (Information Ratio (Russo and Van Roy, 2014))**.** The **information ratio** of a sampling distribution π at time t is:

$$\Psi_t(\pi) := \frac{\Delta_t(\pi)^2}{g_t(\pi)}. \tag{8}$$

This measures the "cost" per bit of information: how much squared regret the agent pays for each unit of information gained about A^*.

IDS selects the distribution π_t^{IDS} that minimizes this ratio. We call $\Psi_t^* = \min_\pi \Psi_t(\pi) = \Psi_t(\pi_t^{\mathrm{IDS}})$ the **minimal information ratio**.

Algorithm 7 Information-Directed Sampling (IDS) (Russo and Van Roy, 2014)

Input: Prior over θ, action set $\mathcal{A}$.
for $t = 1, 2, \ldots, T$ **do**

Compute $\Delta_t(a)$ and $g_t(a)$ for each $a \in \mathcal{A}$.
Solve: $\pi_t^{\text{IDS}} \in \arg\min_{\pi \in \mathcal{D}(\mathcal{A})} \ \Psi_t(\pi) = \dfrac{\Delta_t(\pi)^2}{g_t(\pi)}$.
Sample $a_t \sim \pi_t^{\text{IDS}}$.
Observe outcome, update posterior.

6.2 Connection to Bayesian experimental design

The information gain objective has parallels to **Bayesian Optimal Experimental Design** (BOED). In BOED, a scientist chooses experiments (designs $\xi_1, \xi_2, \ldots$) to learn about an unknown parameter θ as quickly as possible. At each step, the scientist picks a design ξ_{t+1}, observes an outcome $y_{t+1} \sim p(y \mid \xi_{t+1}, \theta)$, and updates a posterior over θ. Think about this like adaptive randomized control trials (though there are some nuanced differences). The reward for each experiment then becomes the reduction in posterior entropy:

$$r_t = H[p(\theta \mid h_t)] - H[p(\theta \mid h_{t+1})], \tag{9}$$

where h_t is the history of designs and outcomes so far.

Notice, though, that this can be thought of as a BAMDP (Section 5.1) where the "environment dynamics" are the experimental model $p(y \mid \xi, \theta)$, the state is the epistemic state $\hat{\theta}$, the actions are designs ξ, and the reward is information gain. Sequential BOED and Bayesian RL can be thought of as similar approaches but coming from different communities and perspectives (Foster et al., 2021a).

6.3 Open question: quantifying information gain

Discussion

But how do you quantify information gain? What about in large action spaces? What about for language models? Let's brainstorm together.

7 Wrap up

You now have a basic idea

References

Agrawal, S. and Goyal, N. (2012). Analysis of Thompson sampling for the multi-armed bandit problem. In *Conference on Learning Theory* (*COLT*), pages 39.1–39.26. JMLR Workshop and Conference Proceedings.

Arumugam, D. and Griffiths, T. L. (2025). Toward efficient exploration by large language model agents. In *EXAIT Workshop at ICML 2025*. arXiv:2504.20997.

Bartók, G., Foster, D. P., Pál, D., Rakhlin, A., and Szepesvári, C. (2014). Partial monitoring — classification, regret bounds, and algorithms. *Mathematics of Operations Research*, 39(4):967–997.

Chapelle, O. and Li, L. (2011). An empirical evaluation of Thompson sampling. In *Advances in Neural Information Processing Systems* (*NeurIPS*), volume 24.

Duff, M. O. (2002). *Optimal Learning: Computational Procedures for Bayes-Adaptive Markov Decision Processes*. PhD thesis, University of Massachusetts Amherst.

Dwaracherla, V., Lu, X., Ibrahimi, M., Osband, I., Wen, Z., and Van Roy, B. (2020). Hypermodels for exploration. In *International Conference on Learning Representations* (*ICLR*).

Dwaracherla, V., Asghari, S. M., Hao, B., and Van Roy, B. (2024). Efficient exploration for LLMs. In *International Conference on Machine Learning* (*ICML*), pages 12215–12227. PMLR.

Eysenbach, B., Gupta, A., Ibarz, J., and Levine, S. (2018). Diversity is all you need: Learning skills without a reward function. *arXiv preprint arXiv:1802.06070*.

Foster, A., Ivanova, D. R., Malik, I., and Rainforth, T. (2021). Deep adaptive design: Amortizing sequential Bayesian experimental design. In *International Conference on Machine Learning* (*ICML*), pages 3384–3395. PMLR.

Foster, D. J., Kakade, S. M., Qian, J., and Rakhlin, A. (2021). The statistical complexity of interactive decision making. *arXiv preprint arXiv:2112.13487*.

Gal, Y. and Ghahramani, Z. (2016). Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning* (*ICML*), pages 1050–1059. PMLR.

Ghafouri, N., Vardakas, J. S., Ramantas, K., and Verikoukis, C. (2024). Energy-efficient intra-domain network slicing for multi-layer orchestration in intelligent-driven distributed 6G networks. *arXiv preprint arXiv:2410.23161*.

Gregor, K., Rezende, D. J., and Wierstra, D. (2016). Variational intrinsic control. *arXiv preprint arXiv:1611.07507*.

Hao, B. and Lattimore, T. (2022). Regret bounds for information-directed reinforcement learning. In *Advances in Neural Information Processing Systems* (*NeurIPS*), volume 35.

Houthooft, R., Chen, X., Duan, Y., Schulman, J., De Turck, F., and Abbeel, P. (2016). VIME: Variational information maximizing exploration. In *Advances in Neural Information Processing Systems* (*NeurIPS*), volume 29.

Jaksch, T., Ortner, R., and Auer, P. (2010). Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 11:1563–1600.

Kaufmann, E., Korda, N., and Munos, R. (2012). Thompson sampling: An asymptotically optimal finite-time analysis. In *International Conference on Algorithmic Learning Theory*, pages 199–213. Springer.

Lai, T. L. and Robbins, H. (1985). Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6(1):4–22.

Lu, X., Van Roy, B., Dwaracherla, V., Ibrahimi, M., Osband, I., and Wen, Z. (2023). Reinforcement learning, bit by bit. *Foundations and Trends in Machine Learning*, 16(6):733–865.

Osband, I., Van Roy, B., Russo, D. J., and Wen, Z. (2019). Deep exploration via randomized value functions. *Journal of Machine Learning Research*, 20(124):1–62.

Lattimore, T. and Györfi, A. (2021). Mirror descent and the information ratio. In *Conference on Learning Theory* (*COLT*), pages 2965–2992. PMLR.

Liu, Z., Chen, C., and co-authors (2025). Sample-efficient alignment for LLMs. In *International Conference on Learning Representations* (*ICLR*).

Osband, I., Russo, D., and Van Roy, B. (2013). (More) efficient reinforcement learning via posterior sampling. In *Advances in Neural Information Processing Systems* (*NeurIPS*), volume 26.

Osband, I., Blundell, C., Pritzel, A., and Van Roy, B. (2016). Deep exploration via bootstrapped DQN. In *Advances in Neural Information Processing Systems* (*NeurIPS*), volume 29.

Osband, I. and Van Roy, B. (2017). Why is posterior sampling better than optimism for reinforcement learning? In *International Conference on Machine Learning* (*ICML*), pages 2701–2710. PMLR.

Osband, I., Aslanides, J., and Cassirer, A. (2018). Randomized prior functions for deep reinforcement learning. In *Advances in Neural Information Processing Systems* (*NeurIPS*), volume 31.

Osband, I., Wen, Z., Asghari, S. M., Dwaracherla, V., Ibrahimi, M., Lu, X., and Van Roy, B. (2023). Epistemic neural networks. In *Advances in Neural Information Processing Systems* (*NeurIPS*), volume 36.

Russo, D. and Van Roy, B. (2014). Learning to optimize via information-directed sampling. In *Advances in Neural Information Processing Systems* (*NeurIPS*), volume 27.

Russo, D. and Van Roy, B. (2016). An information-theoretic analysis of Thompson sampling. *Journal of Machine Learning Research*, 17(68):1–30.

Russo, D. and Van Roy, B. (2018). Learning to optimize via information-directed sampling. *Operations Research*, 66(1):230–252.

Russo, D., Van Roy, B., Kazerouni, A., Osband, I., and Wen, Z. (2018). A tutorial on Thompson sampling. *Foundations and Trends in Machine Learning*, 11(1):1–96.

Strens, M. (2000). A Bayesian framework for reinforcement learning. In *International Conference on Machine Learning* (*ICML*), pages 943–950.

Thompson, W. R. (1933). On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294.

Zimmert, J. and Lattimore, T. (2019). Connections between mirror descent, Thompson sampling and the information ratio. In *Advances in Neural Information Processing Systems* (*NeurIPS*), volume 32.