

Problem set 3: Multi-armed bandits

Foundations of Machine Learning, Hilary term 2027

Maximilian Kasy

What this problem set is about

In a bandit problem, we choose treatments for a sequence of units, and learn about the effects of the treatments as we go along. Every choice involves a tradeoff between *exploitation* (choosing the treatment that looks best now) and *exploration* (learning more, so that we choose better later). In lectures, we discussed Thompson sampling, a simple heuristic for this problem, and bounds on its regret, both when the parameters are held fixed and when treatment effects are small relative to the sample size.

In this problem set, you will simulate Thompson sampling and see these regret bounds at work (Problem 1). You will then compare Thompson sampling to exploration sampling, which is designed for a different goal: finding the best treatment at the end of an experiment, rather than treating the participants of the experiment well (Problem 2). Problem 2 is part of the coding portfolio, but it is **not examined**: the exam will not ask about exploration sampling or policy regret.

How this problem set works

1. **Software.** This problem set can be solved in Python or R.
2. **Your report.** Your code should run from start to end in one execution, producing all the output. Code, output, and discussion should be integrated in a single report (a Jupyter notebook, or R-Markdown/Quarto), which you submit as a pdf. Please label all figures and tables clearly, so that someone who has not read your code can understand them.
3. **Speed.** We need many replications of each experiment. I strongly recommend simulating all R replications at the same time (“vectorized” code, which stores the posterior parameters of all replications in an $R \times k$ array and updates them all at once), rather than looping over replications. With vectorized code, my solution runs in about five minutes on a laptop. Parallel computing (for instance the *future* package in R) is optional.
4. **Predictions.** Throughout the problem set, you will find blue boxes labelled *Predict* asking you to **predict** what you will find, *before* you run the corresponding code. Please take these seriously: they are the most useful part of the exercise. Write down your prediction in your report, in a few sentences, together with a brief reason. Keep your prediction in the report even if it turns out to be wrong; your predictions are not assessed on whether they are correct. At the end of each problem, a short section asks

you to look back at your predictions: Were they right? If not, what did you overlook? This is the discussion part of your report; there is no separate discussion to write.

5. **Checkpoints.** The grey boxes list numbered *checkpoints*, [C3.1], [C3.2], and so on: specific numbers that your code should produce. For some of them, the theory from class tells you what the value should be; in that case, the box says so. If your number differs from the theoretical prediction by more than Monte Carlo error, say so in your report and try to explain why. At the end of your report, print all checkpoints in one block, one per line, in exactly this format:

CHECKPOINT C3.1 = 0.3000

for instance using

```
print(f"CHECKPOINT C3.1 = {value:.4f}")      # Python
cat(sprintf("CHECKPOINT C3.1 = %.4f\n", value)) # R
```

6. **Preparing for the exam.** Problem 1 ends with a green box of *questions for the exam*. You do not need to answer these in your report. The exam will contain questions similar to these, about figures like the ones you produce here, and about short pieces of code from my solution. If you have done the problem set yourself and thought about your predictions, you should find them straightforward.

FROM THE LECTURES the Bernoulli bandit and Thompson sampling

There are k treatments (“arms”). In each period $t = 1, \dots, T$, one unit arrives, we assign a treatment $D_t \in \{1, \dots, k\}$, and observe a binary outcome,

$$Y_t = Y_t^{D_t}, \quad Y_t^d \sim \text{Ber}(\theta^d).$$

The best arm has success probability $\theta^* = \max_d \theta^d$, and choosing arm d instead costs $\Delta^d = \theta^* - \theta^d$. The **cumulative average regret** after T periods is

$$R_T = E \left[\frac{1}{T} \sum_{t=1}^T \Delta^{D_t} \right].$$

Posterior. We use a uniform prior, $(\theta^1, \dots, \theta^k) \sim U([0, 1]^k)$. After period t , the posterior for θ^d is $\text{Beta}(\alpha_t^d, \beta_t^d)$, where $\alpha_t^d = 1 + (\text{number of successes for arm } d \text{ so far})$ and $\beta_t^d = 1 + (\text{number of failures for arm } d \text{ so far})$.

Thompson sampling. In each period, draw one value $\hat{\theta}_t^d$ from the current posterior of each arm, and assign the arm with the largest draw, $D_t = \arg\max_d \hat{\theta}_t^d$. Equivalently: assign each arm with probability equal to the posterior probability that it is the best arm.

Problem 1: Regret of Thompson sampling

(a) Simulation functions

Write two functions, which you will use throughout the problem set.

1. `bandit_history(T, theta, R, algorithm)` simulates R independent replications of a bandit experiment, with T periods, success probabilities $\theta = (\theta^1, \dots, \theta^k)$, and treatment assignment by `algorithm` (for now, only Thompson sampling). It returns the

$R \times T$ arrays of treatments D_t and outcomes Y_t , and, for each replication, the arm d_T^* with the highest posterior mean at the end of the experiment.

2. `summarize(T, theta, R, algorithm)` calls `bandit_history` and summarizes the results. For each period t , it returns the average across replications of Y_t , of θ^{D_t} , of $\mathbf{1}(D_t \text{ is the best arm})$, of Δ^{D_t} , and of the running average $\frac{1}{t} \sum_{s \leq t} \Delta^{D_s}$ (whose average across replications is R_t). It also returns the average across replications of the *policy regret* $\theta^* - \theta^{d_T^*}$, and the share of replications in which d_T^* is the best arm.

(b) Regret over time, for a fixed parameter

Let $\theta = (0.2, 0.5, 0.8)$, $T = 1000$, and $R = 10,000$.

PREDICT P3.1

before you run the code

- What is R_1 , the expected regret in the very first period? (Think about what Thompson sampling does before it has seen any data.)
- How quickly will R_T go to zero as T grows? Recall the rate we derived in class.
- Which of the two suboptimal arms will Thompson sampling have more trouble ruling out?

WHAT TO REPORT

Figure 3.1: Two panels.

- (i) R_T as a function of T .
- (ii) $T \cdot R_T / \log(T)$ as a function of T (with T on a log scale), and a horizontal line at the constant in the Lai-Robbins bound from the slides, $\sum_{d: \Delta^d > 0} \Delta^d / kl(\theta^d, \theta^*)$, where $kl(p, q) = p \log(p/q) + (1-p) \log((1-p)/(1-q))$.

[C3.1]: R_1 . Theory: 0.3.

[C3.2]: R_{100} . [C3.3]: R_{1000} .

[C3.4]: the Lai-Robbins constant (this involves no randomness).

(c) Small differences between the arms

The regret rate in part (b) holds for a fixed θ , as T grows. In class, we saw that things look different when the arms are hard to tell apart, relative to the sample size. To see this, consider two arms, with $\theta^1 = 0.5$ fixed, $T = 200$, $R = 10,000$, and vary θ^2 over the grid $\{0, 0.05, 0.1, \dots, 1\}$.

PREDICT P3.2

before you run the code

- What will R_{200} be when $\theta^2 = 0.5$? What about when $\theta^2 = 1$: large or small?
- For roughly which values of θ^2 do you expect regret to be largest?

WHAT TO REPORT

Figure 3.2: Two panels, both with θ^2 on the horizontal axis. (i) R_{200} . (ii) The share of all assignments (across all periods and replications) that went to the better arm.

[C3.5]: the largest value of R_{200} across the grid of values for θ^2 .

[C3.6]: the distance $|\theta^2 - \theta^1|$ at which this largest value is attained. (The curve is roughly symmetric around $\theta^2 = 0.5$, so the largest value may be on either side.)

(d) Looking back at your predictions

Return to your predictions P3.1 and P3.2. For each one, say in a sentence or two whether it was right, and if not, what you overlooked. Where you can, relate your answer to the regret bounds discussed in class, and to Figure 3 in Wager and Xu (2021), which is shown in the slides.

QUESTIONS FOR THE EXAM

1. Explain why $R_1 = 0.3$ for $\theta = (0.2, 0.5, 0.8)$. What would R_1 be for $\theta = (0.1, 0.9)$?
2. Write down the posterior for each arm after the following data: arm 1 was assigned 4 times with 3 successes, arm 2 was assigned twice with no successes. In your code, which lines perform this update?
3. In class, we showed that the regret of good bandit algorithms goes to zero at the rate $\log(T)/T$ when θ is fixed. Is Figure 3.1 consistent with this?
4. For $\theta = (0.2, 0.5, 0.8)$, which of the two suboptimal arms contributes more to the Lai-Robbins constant? Why does this make sense?
5. In Figure 3.2, why is R_{200} zero at $\theta^2 = 0.5$, and small for θ^2 close to 1? Why is it largest in between? Relate your answer to the heuristic calculation in the slides, where regret is of order $\Delta \cdot \Phi(-2\sqrt{T}\Delta)$.
6. Based on this heuristic, how would you expect the peaks in Figure 3.2 to move if $T = 800$ instead of $T = 200$?

Problem 2: Exploration sampling (not examined)

NOT EXAMINED

This problem is part of your coding portfolio, and you need to complete it and report its checkpoints like the others. It is **not examined**: it builds on the bonus slides on *Experiments for policy choice*, which go beyond the main lectures, and the exam will not ask about it.

The idea

Thompson sampling is designed to treat the participants in the experiment well: it quickly shifts most units to the arm that looks best. Often, however, the purpose of an experiment is to find the best treatment for a policy that is implemented *after* the experiment. For this purpose, what matters is the **policy regret** $\theta^* - \theta^{d_T^*}$ of the arm d_T^* that we choose at the end, rather than the **in-sample regret** R_T .

Exploration sampling (Kasy and Sautmann, 2021) is a modification of Thompson sampling designed for this goal, discussed in the bonus slides on *Experiments for policy choice*. Let p_t^d be the posterior probability, given the data from periods $1, \dots, t-1$, that arm d is the best arm. (Thompson sampling assigns arm d with probability p_t^d .) Exploration sampling instead assigns arm d with probability

$$q_t^d = \frac{p_t^d(1 - p_t^d)}{\sum_{d'} p_t^{d'}(1 - p_t^{d'})}.$$

Computing p_t^d . Unlike Thompson sampling, exploration sampling needs the probabilities p_t^d themselves, for each replication and period, and they need to be accurate. You can approximate p_t^d by Monte Carlo, using at least 1,000 posterior draws per replication and period.

(a) Simulation functions

Extend your functions `bandit_history` and `summarize` from Problem 1, so that `algorithm` can be either Thompson sampling or exploration sampling.

(b) Comparison

Let $T = 200$ and $R = 10,000$. Consider two parameter vectors: an “easy” one, $\theta = (0.2, 0.5, 0.8)$, where the best arm is easy to find, and a “hard” one, $\theta = (0.5, 0.55, 0.6)$.

PREDICT P3.3

before you run the code

- Which algorithm will have lower in-sample regret R_{200} ? Which will have lower policy regret?
- For the easy parameter vector, the posterior quickly becomes confident about which arm is best. What share of units will each algorithm assign to the best arm in the last periods?

For both parameter vectors and both algorithms, compute the in-sample regret R_{200} , the share of assignments to the best arm, the policy regret, and the probability of choosing the best arm at the end.

WHAT TO REPORT

Table 3.1: These four statistics, for each of the four combinations of parameter vector and algorithm.

Figure 3.3: Two panels, one for each parameter vector. Horizontal axis: period t . Vertical axis: share of replications assigned to the best arm in period t . One line for each algorithm.

[C3.7]: R_{200} , Thompson sampling, easy.

[C3.8]: R_{200} , exploration sampling, easy.

[C3.9]: share of replications assigned to the best arm in period $t = 200$, exploration sampling, easy.

[C3.10]: policy regret, Thompson sampling, hard.

[C3.11]: policy regret, exploration sampling, hard.

[C3.12]: probability of choosing the best arm, Thompson sampling, hard.

[C3.13]: probability of choosing the best arm, exploration sampling, hard.

(c) Looking back at your predictions

Return to your prediction P3.3. For each part, say in a sentence or two whether it was right, and if not, what you overlooked. Do your results line up with the simulation findings summarized in the slides and in Kasy and Sautmann (2021)?

QUESTIONS TO THINK ABOUT (NOT EXAMINED)

1. Define in-sample regret and policy regret. Give an example of an experiment where each would be the right objective.
2. Suppose $p_t = (0.1, 0.3, 0.6)$. Compute the assignment probabilities of Thompson sampling and of exploration sampling. Which arm loses the most under exploration sampling, relative to Thompson sampling?
3. In Table 3.1, which algorithm does better on in-sample regret, and which on policy regret? Is this consistent with the slides?
4. Use Figure 3.3 to explain why exploration sampling has much higher in-sample regret than Thompson sampling for the easy parameter vector.

References

- Kasy, M. and Sautmann, A. (2021). Adaptive treatment assignment in experiments for policy choice. *Econometrica*, 89(1):113–132.
- Wager, S. and Xu, K. (2021). Diffusion asymptotics for sequential experiments. *arXiv preprint arXiv:2101.09855*.