ECE 8803: Online Decision Making in Machine Learning **Fall 2021**

Lecture 16: October 20

Lecturer: Vidya Muthukumar

Last lecture, we introduced Bayesian reasoning in the K-armed bandit problem. Instead of treating the mean rewards $\mu_1, \dots, \mu_K$ of the K arms as entirely unknown, Bayesian reasoning incorporates a *prior* belief over these means, and uses the prior together with observations of the rewards to construct *posterior beliefs* about the means. We worked through a few examples of computation of these posterior beliefs in three cases: a) a uniform (or uninformative) prior, b) an informative (well-specified) prior, and c) an ill-judged (misspecified) prior.

We then introduced the Thompson sampling algorithm to decide how to take actions as a function of these posterior beliefs at any round of decision-making. For completeness, the Thompson sampling algorithm is detailed here again in Algorithm 1.

Algorithm 1 Thompson sampling algorithm

1: **Input:** Prior $\boldsymbol{Q}^{(a)}$ on arm a for $a = 1, \dots, K$
2: **for** $t = 1, \dots, T$ **do**
3: Compute posterior distribution $\boldsymbol{Q}_t^{(a)}$ on μ_a from observed samples
4: Sample $(\mu_{1,t}, \mu_{2,t}, \dots, \mu_{K,t})$ from the posterior distributions $\boldsymbol{Q}_t^{(a)}$
5: Pull arm $A_t = \arg\max_{a \in \{1,\dots,K\}} \mu_{a,t}$, and observe reward G_{t,A_t}.
6: **end for**

Step 4 of *sampling* from the posterior is what makes Thompson sampling a randomized algorithm. Last class, we saw the positive ramifications of this type of randomization in particular for the case of an ill-judged prior: although the posterior from a few samples was tilting towards the wrong direction (picking the suboptimal arm), there was still a sizable probability that the optimal arm would be picked, and sufficient data would be collected to overcome the effect of the ill-judged prior. We also saw that Thompson sampling seems to automatically reduce its level of randomization when an informative prior is chosen that is consistent with the true reward distribution. Today, we will explore quantitatively how this benefit manifests.

Finally, we touched upon implementation issues of Algorithm 1. The most computationally complex steps are Steps 3 and 4 of computing the posterior distribution and sampling from it. We saw last class that the choice of a Beta prior, together with Bernoulli rewards, ensures that the posterior is also Beta-distributed and can be easily computed via updating the parameters of the Beta distribution. This leads to a very convenient and efficient implementation of Thompson sampling. Such prior-reward distribu-

tion pairs are commonly called *conjugate* priors in the Bayesian statistics literature, and there are several examples other than the Beta-Bernoulli pair. For more examples, see: `https://en.wikipedia.org/wiki/Conjugate_prior`. Steps 3 and 4 are in general difficult in the absence of such conjugate prior structure—however, recent advances in geometric methods in optimization and sampling[1] do open the door to the possibility of sampling from the posterior without using an exact expression for it, i.e. bypassing Step 3 and moving to Step 4 directly. In other cases, we can use graphical models to model the random reward distribution and execute Step 3 efficiently via powerful iterative algorithms. These techniques greatly increase the potential of algorithms like Thompson sampling to be applied very broadly. (While these topics are not the focus of this course, they are fascinating and well worth learning more about!)

Today, we will briefly overview the guarantees that Thompson sampling gives us on pseudo-regret. We will be particularly interested in pseudo-regret bounds that illustrate the benefits of Thompson sampling over UCB when we have an informative prior for each arm, or prior knowledge of structure across arms. Towards the end of the lecture, we will also touch upon a different type of randomized algorithm that works with non-stochastic (adversarial!) rewards.

16.1. Pseudo-regret of Thompson sampling

For simplicity, we will continue to assume that the means lie between 0 and 1, i.e. $\mu_i \in [0,1]$. A starting question is, whether Thompson sampling achieves logarithmic pseudo-regret guarantees of the form that UCB did for an arbitrary instance $(\mu_1, \ldots, \mu_K)$, i.e. that scale inversely in the *gaps* between the optimal arm and suboptimal arms' mean rewards.

Of course, to make this question well-posed we need to specify a prior for the Thompson sampling algorithm. It turns out to be particularly convenient to consider the Beta prior with parameters $(1,1)$ for the mean of each arm μ_i (and the distribution is assumed to be independent across the arms). Recall from the previous lecture that in this case, the Beta prior basically becomes the uniform prior, i.e. the uniform distribution on $[0,1]$. This represents a situation in which we are learning from scratch. It turns out that under this situation, we can get pseudo-regret upper bounds on Thompson sampling that are basically identical to those of UCB, as provided below.

Theorem 1 (Kaufmann et al. (2012); Agrawal and Goyal (2017)) *Thompson sampling with the Beta*$(1,1)$ *prior and mean rewards* $\boldsymbol{\mu} := (\mu_1, \ldots, \mu_K)$*, where* $\mu_i \in [0,1]$*, achieves pseudo-regret given by*

$$\overline{R}_T(\boldsymbol{\mu}) = \mathcal{O}\left((1+\epsilon)\sum_{a \neq a^*} \frac{\log T}{\Delta_a} + \frac{K}{\epsilon^2}\right)$$

for any value of $\epsilon \in (0,1)$*. For example, substituting* $\epsilon = 1/2$ *gives us*

$$\overline{R}_T(\boldsymbol{\mu}) = \mathcal{O}\left(\sum_{a \neq a^*} \frac{\log T}{\Delta_a} + K\right)$$

1. `https://simons.berkeley.edu/programs/gmos2021`

Theorem 1 shows that Thompson sampling achieves gap-dependent pseudo-regret bounds much in the same way as UCB. This is a nice sanity check that Thompson sampling is indeed a good algorithm to use, even when we don't have particularly useful prior information—in terms of its dependence on the number of rounds T and the gaps $\{\Delta_a\}_{a \neq a^*}$, it has comparable performance. It is worth recalling here that Thompson sampling was introcued as a heuristic in 1933. These bounds were first provided in 2012-2013—it is quite remarkable that this "heuristic" turns out to achieve near-optimal performance!

16.2. Exploiting prior information: *Bayesian* pseudo-regret

While a nice sanity check, Theorem 1 does not explain why we may actually prefer to use Thompson sampling over algorithms that learn from scratch like UCB. This is because it only applies out-of-the-box to the "uniform prior" case. Remember that our real motivation for using a Bayesian algorithm like Thompson sampling is to exploit useful prior information that we may have about the reward distributions. The hope is that the more informative the prior is, the more the pseudo-regret is driven down to be lower (as we have less to learn).

We now present this type of guarantee on Thompson sampling which will reflect the benefits of an informative prior. To state this guarantee, we first need to define a Bayesian notion of pseudo-regret.

Definition 2 (Bayesian pseudo-regret) *Consider a prior distribution $\boldsymbol{Q}(\cdot)$ on the mean rewards $\boldsymbol{\mu} := (\mu_1, \ldots, \mu_K)$. Then, the **Bayesian** pseudo-regret is defined as $\mathbb{E}_{\boldsymbol{Q}(\cdot)}\left[\overline{R}_T(\boldsymbol{\mu})\right]$, i.e. we consider the expected pseudo-regret over the prior distribution on the instances. Note that the Bayesian pseudo-regret is a function of the algorithm chosen and the prior distribution $\boldsymbol{Q}(\cdot)$.*

It is instructive to carry over the discrete-prior example from last lecture to get a better window into the Bayesian pseudo-regret and what it means.

Example 1 (Discrete prior) *Our discrete prior example considered $\boldsymbol{\mu} = (\mu_A, \mu_B)$ and a prior distribution over two values for $\boldsymbol{\mu}$: $(0.6, 0.4)$ and $(0.2, 0.4)$. Thus, the Bayesian pseudo-regret is given by*

$$Q((0.6, 0.4)) \cdot \overline{R}_T((0.6, 0.4)) + Q((0.2, 0.4)) \cdot \overline{R}_T((0.2, 0.4)).$$

In the uniform-prior case, we had $Q((0.6, 0.4)) = 0.5$ and so the Bayesian pseudo-regret would equally weight the pseudo-regret of an algorithm on the instance where Drug A is optimal and the instance where Drug B is optimal. In the informative-prior case, we had $Q((0.6, 0.4)) = 0.8$ and so the Bayesian pseudo-regret heavily upweights the pseudo-regret of an algorithm on the instance where Drug A is optimal—thus, having prior information about this possibility would intuitively be helpful. We will see this in a precise quantitative way in a few moments.

Example 2 (Beta prior) *In the Beta prior example, we considered $\mu_B = 0.4$ to be fixed (as in the preceding discrete prior example) and $\boldsymbol{Q}(\mu_A) = Beta(\alpha, \beta)$ for parameters α, β. In this case, the Bayesian pseudo-regret is given by*

$$\mathbb{E}_{\boldsymbol{Q}(\mu_A)}\left[\overline{R}_T((\mu_A, 0.4))\right].$$

While calculating this expectation is now significantly more complicated, the same qualitative ideas hold. For the case $\alpha = \beta = 1$, the Beta prior is the uniform prior and we can think of the Bayesian pseudo-regret as measuring the average-case pseudo-regret of an algorithm over a uniform distribution on all possible bandit instances. On the other hand, if $\alpha \gg \beta$, the Beta prior is heavily weighted towards instances that favor Drug A over B, and so the Bayesian pseudo-regret will overweight these instances as well and underweight others.

Examples 1 and 2 hint at the fact that an informative prior, when chosen correctly, can really reduce the Bayesian pseudo-regret and thus speed up performance. Think about an extreme scenario: suppose that our prior said that Drug A was better than Drug B *with probability* 1, and we also evaluated Bayesian pseudo-regret under this prior. Then, this maximally informative prior enables us to never pick the suboptimal Drug B; moreover, the Bayesian pseudo-regret only evaluates instances where Drug B is suboptimal. Therefore, in this extreme case we would get Bayesian pseudo-regret equal to *zero*!

This intuition turns out to be true in a more continuous sense: the more "informative" the prior, the lower it is possible to make the Bayesian pseudo-regret. We now define such a quantity that is smaller when the prior is more informative—this quantity is one that you encountered in HW 2, and is called the *entropy function*. We define it below.

Definition 3 *For a prior distribution $\boldsymbol{Q}(\cdot)$ on the means $(\mu_1, \dots, \mu_K)$, we can define the probability that a is the optimal arm, i.e. for each $a \in \{1, \dots, K\}$, we can define $P_a := \boldsymbol{Q}(\mu_a > \mu_{a'} \text{ for all } a' \neq a)$. Then, $(P_1, \dots, P_K)$ defines a valid probability distribution, and we can write the entropy function of this "optimal action" distribution as*

$$H_{\boldsymbol{Q}}(A^*) := -\sum_{i=1}^{K} P_i \log(P_i).$$

We will call this the "optimal-action entropy" for shorthand henceforth.

The more "informative" a prior distribution, the smaller this *optimal-action entropy* will be. In the discrete prior example, we get $H_{\boldsymbol{Q}(A^*)} = H_{\mathsf{binary}}(Q((0.6, 0.4))$, where $H_{\mathsf{binary}}(\cdot)$ denotes the binary entropy function. This function is plotted on Figure 16.1. You can clearly see that the informative prior $Q((0.6, 0.4)) = 0.8$ has much lower optimal-action entropy than the uninformative uniform prior $Q((0.6, 0.4)) = 0.5$.

With all of this notation set up, we are now ready to state a guarantee on the Bayesian pseudo-regret on Thompson sampling that is smaller when our prior provides more useful information.

Theorem 4 (Russo and Van Roy (2016)) *Thompson sampling **with any prior $\boldsymbol{Q}$** and bounded rewards in $[0, 1]$ achieves Bayesian regret*

$$\mathbb{E}_{\boldsymbol{Q}(\cdot)}\left[\overline{R}_T(\boldsymbol{\mu})\right] = \mathcal{O}\left(\sqrt{\Gamma T \cdot H_{\boldsymbol{Q}}(A^*)}\right)$$

where Γ is a measure of the number of degrees of freedom in the problem (worst case equal to K).

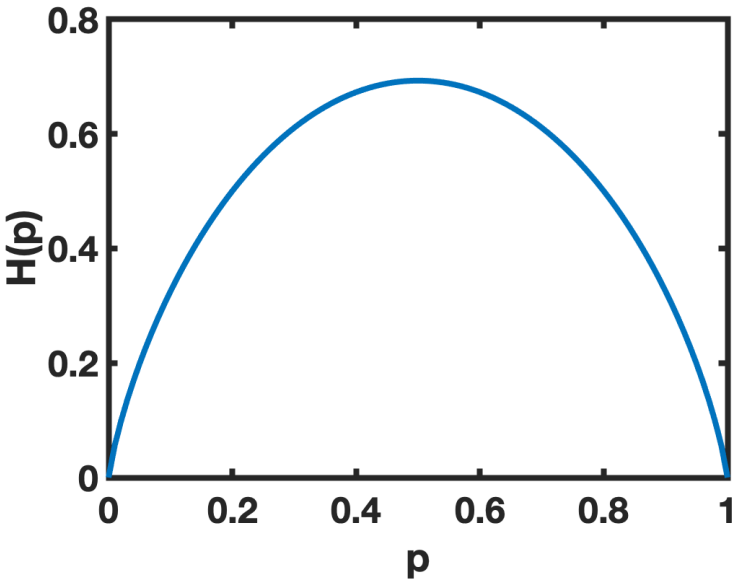

Figure 16.1: The binary entropy function.

Theorem 4 is not entirely comparable to the kind of result in Theorem 1: the dependence on the number of rounds T is clearly worse. However, this result provides a quantitative explanation for why having more prior information is useful: it drives the optimal-action entropy down, and therefore drives pseudo-regret down.

Precisely understanding the Bayesian regret of Thompson sampling (with upper bounds and lower bounds that match, and are logarithmic in T) remains an open and challenging research direction. Moreover, there is a critical blanket assumption made in the definition of Bayesian regret that *the prior chosen by Thompson sampling exactly matches the prior given by Nature* (as the latter is what is used to evaluate the Bayesian pseudo-regret). It is not clear whether this will be the case in practice; Theorem 4 does not at all apply if the prior is ill-judged (i.e. Thompson sampling chooses prior $\boldsymbol{Q}$, but Bayesian regret is evaluated with respect to a different prior $\boldsymbol{Q}' \neq \boldsymbol{Q}$). All of these points make the discussion about whether to use Thompson sampling or UCB in practice an extremely nuanced and open research direction—the answer is often dependent on context, modeling assumptions, and how much prior information we have and how reliable that prior information is. Given how much remains to be understood about the fine-grained behavior of Thompson sampling in particular, it is also often a matter of taste!

16.2.1 Other benefits: structure across arms

We conclude our discussion about Thompson sampling by providing and informally describing a second example that is different in nature from drug discovery, and will involve *strong correlations* in reward distributions across a very large number of arms that we would like to exploit to drive down the linear dependence on the total number of arms. This example

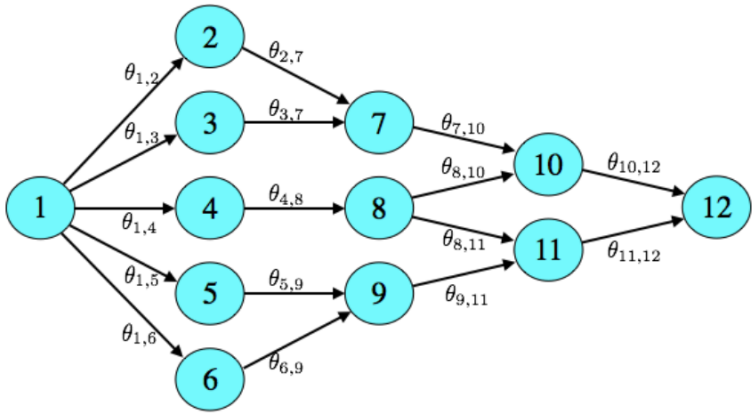

Figure 16.2: A depiction of the shortest-path problem with 12 vertices. Here, vertex 1 is the source vertex and 12 is the target (destination) vertex.

illustrates the versatility of the Thompson sampling algorithm under diverse modeling assumptions.

Example 3 (Shortest path on a graph) *This example is Example 1.2 in Russo et al. (2018) and is also discussed at length in Chapters 4 and 8. Suppose that we have moved to a new city, and we are commuting from home to work every morning*[2]*. Then, we would want to pick the path that requires the least expected travel time; however, we are uncertain about the travel time along different routes and we need to explore to figure this shortest path out.*

This constitutes an online-shortest-path problem on a graph, and the vertices represent intermediate stopping points (e.g. intersections, or stop signs). As Figure 16.2 demonstrates, the mean travel time along an edge e is denoted by θ_e, and a path $\boldsymbol{e} := (e_1, \ldots, e_n)$ from source s to destination t will lead to total travel time $\sum_{i=1}^{n} \theta_{e_i}$. Our goal is to minimize pseudo-regret with respect to the best path, i.e. the one that minimizes the total travel time. We denote this path by $\boldsymbol{e}^* := (e_1^*, \ldots, e_n^*)$.

This is, of course, an example of the multi-armed bandit problem (where the "arms" represent all possible paths in the graph). Unfortunately, at first glance it seems to be very large in scale: the number of total paths can scale **exponentially** in the number of vertices/edges, and since we have seen that pseudo-regret tends to increase with the number of arms in a linear fashion, this could lead to a very suboptimal guarantee. Moreover, a UCB-type algorithm in its naive form would also need to enumerate all possible paths, which is also computationally expensive for the same reason of exponential scaling. Can we do better?

The central idea that can be exploited is that at every round, we observe not only the total travel time, but also the travel time along each of the edge segments of the path that we tried. Moreover, several edges are common to multiple paths, so observing the travel time on a particular edge will tell us about the travel time of all paths that use that

2. and we have decided to experiment on our own instead of placing our trust in Google Maps.

edge. It turns out that Thompson sampling can be adapted to exploit this in a nice way: instead of simply computing posterior distributions on the total travel times $\sum_{i=1}^{n} \theta_{e_i}$, we can compute posterior distributions on the individual travel times θ_e, sample from them, and pick an optimal path (which is itself a linear-time operation in the number of vertices and edges). Not only does this lead to a more efficient algorithm naturally, but it also leads to a vastly improved pseudo-regret guarantee! In fact, Theorem 4 can be shown to yield $\mathbb{E}_{\boldsymbol{Q}}[\overline{R}_T] = \mathcal{O}(\sqrt{|E| H_{\boldsymbol{Q}}(A^*) T}$, where $|E|$ denotes the number of edges in the graph. This is *much* better than the $\mathcal{O}(2^{|E|})$ guarantee that would be obtained via UCB in its naive form[3], and showcases the versatility and utility of the Thompson sampling algorithm in adapting not only to prior structure on a single arm (as in the drug discovery example), but also in shared structure across arms.

16.3. Additional references

- See Chapter 8, Russo et al. (2018) for additional context for how and why the Thompson sampling algorithm works. Much of the discussion on the guarantees presented in this lecture is also provided there. This chapter also discusses certain shortcomings of the Thompson sampling algorithm that can arise either from excessive randomization with an uninformative prior, or too little randomization with an ill-judged prior.

- The first pseudo-regret analysis of Thompson sampling was conducted by Agrawal and Goyal (2012); while this recovered the $\mathcal{O}(\log T)$ dependence, its dependence on the gaps $\{\Delta_a\}$ was highly suboptimal. Theorem 1 is a restatement of the optimal improved analysis Agrawal and Goyal (2017).

- Theorem 4 is due to Russo and Van Roy (2016) and uses an information-theoretic style of analysis. The first and most important step in the analysis is that the prior used by Thompson sampling exactly matches the prior that is given by Nature. This turns out to imply that after t rounds, the posterior used by Thompson sampling also matches the true posterior distribution on the rewards. Whether and when this analysis can be extended to situations in which the posterior does not match remains unclear.

- The Thompson sampling algorithm is very popular, but it is by no means the only bandit algorithm that is well-suited for a Bayesian setting. A Bayesian version of UCB exists Kaufmann et al. (2012), and regular pseudo-regret as well as Bayesian pseudo-regret can be evaluated under it. In fact, the Bayesian pseudo-regret provides a principled metric to evaluate exact optimality of a bandit algorithm (when the prior is chosen suitably, such as uniform). The optimal algorithm turns out to be related to UCB and is called a *Gittins index policy* Gittins (1979). This optimal algorithm was in fact discovered and proved to be optimal well before the discovery of the simpler UCB-based ideas.

3. It is important to point out here that this is not an entirely fair comparison. Variants of UCB that deal better with the combinatorial structure of this problem, and construct individual confidence intervals on the θ_e's, can be devised. However, the Thompson sampling algorithm is especially advantageous in that no modifications need to be made to the original algorithm, and the structure is only implicitly encoded, not explicitly. This leads to great ease of use in practice.

References

Shipra Agrawal and Navin Goyal. Analysis of thompson sampling for the multi-armed bandit problem. In *Conference on learning theory*, pages 39–1. JMLR Workshop and Conference Proceedings, 2012.

Shipra Agrawal and Navin Goyal. Near-optimal regret bounds for thompson sampling. *Journal of the ACM (JACM)*, 64(5):1–24, 2017.

John C Gittins. Bandit processes and dynamic allocation indices. *Journal of the Royal Statistical Society: Series B (Methodological)*, 41(2):148–164, 1979.

Emilie Kaufmann, Nathaniel Korda, and Rémi Munos. Thompson sampling: An asymptotically optimal finite-time analysis. In *International conference on algorithmic learning theory*, pages 199–213. Springer, 2012.

Daniel Russo and Benjamin Van Roy. An information-theoretic analysis of thompson sampling. *The Journal of Machine Learning Research*, 17(1):2442–2471, 2016.

Daniel J Russo, Benjamin Van Roy, Abbas Kazerouni, Ian Osband, and Zheng Wen. A tutorial on thompson sampling. *Foundations and Trends® in Machine Learning*, 11(1): 1–96, 2018.