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

Lecture 15: October 18

Lecturer: Vidya Muthukumar

Last lecture, we completed our discussion of the UCB algorithm, and demonstrated that, in a certain sense, it is the best that we can do for the canonical multi-armed bandit problem. Our reasoning about why we could not do better, however, critically relied on the fact that we were really learning from scratch in every way! To see this, let us recall the example we provided and its high-level properties. We considered our example of drug discovery in which Drug B is *known* to have an average efficacy of 0.4 in treating patients, but Drug A either has a superior efficacy of 0.6, or an inferior efficacy of 0.2—and we do not know which of the situations we are in. To distinguish the two situations, we *need* to sample Drug A a minimal number of times and this leads to a lower bound on the pseudo-regret. It is worth recalling two distinct reasons for why this is *necessary*:

- We have zero prior information about which situation we are in with respect to Drug A's efficacy—it is equally likely to be relatively ineffective, or relatively more effective.

- Sampling the efficacy of Drug B tells us nothing about the efficacy of Drug A, i.e. there is no shared information across the arms.

This paucity of information at the beginning of the learning process reflects how idealized the canonical form of the multi-armed bandit problem really is. In practice, we are never learning entirely from scratch. For example, drug A might be a new drug that has been manufactured by a reputed[1] pharmaceutical company with a track record of previously manufactured drugs that goes one way or another, and so we might have an educated guess about how effective their new drug might be based on that track record. Moreover, Drug B and Drug A may share some common features (think Pfizer and Moderna vaccines!), and so trials from Drug B may give us at least some partial information about whether Drug A is also effective or not.

With the possible presence of such *prior information*, we may no longer wish to deploy an algorithm like UCB that truly learns from scratch. The natural question is whether there are alternative algorithms that behave (in an approximate sense) like UCB when there is no side information, but are able to flexibly incorporate this prior information. In this lecture, we will describe an algorithm that does exactly this, that was introduced in the year 1933 (Thompson, 1933), but only well-understood in the last decade.

1. Note that reputation is not always good. :)

15.1. A Bayesian perspective on multi-armed bandits

To formally model prior information, we will imbue the multi-armed bandit (MAB) problem with a Bayesian framework. We have modeled the K-armed bandit problem as selecting one amongst K arms, where for each $a \in \{1, \ldots, K\}$, arm a has a mean reward of μ_a. The key reason that we need to *learn* in the MAB formulation, and the key reason the exploration-exploitation tradeoff arises in the first place, is because the means $\{\mu_a\}_{a=1}^K$ are *unknown* apriori. So far, we have taken (what is commonly called in statistical methodology) a *frequentist* approach to the problem: we collect samples from each arm (in an adaptive manner), construct estimates of the means, and use these estimates in some way to make future decisions. The frequentist approach works well in a setting where the means $\{\mu_a\}_{a=1}^K$ are completely unknown. However, in many cases we may have reasonable *prior information* on where these means are coming from, and we may want to use that information together with the samples of rewards that we gather in an intelligent way. For our purposes, we define a **prior** $\boldsymbol{Q}$ as a probability distribution on the means $\{\mu_1, \ldots, \mu_K\}$ that we begin the learning process with. The prior represents our belief system about where the mean rewards are coming from, and most often is an input to a Bayesian algorithm.

We should expect a Bayesian algorithm that utilizes the prior distribution $\boldsymbol{Q}$ to work particularly well when the prior *reflects reality* in the following sense: at the very beginning of the learning process, the means $\{\mu_1, \ldots, \mu_K\}$ are themselves actually drawn by Nature from the prior $\boldsymbol{Q}$. Let us consider two concrete examples below, both of which use Bernoulli rewards, to illustrate this idea of a prior and what we can do with it.

15.1.1 Example: Bernoulli rewards

Consider our drug discovery problems with two drugs, Drug A and Drug B. For this section, the "rewards", or efficacy of the drugs, will be modeled as Bernoulli random variables, i.e. taking values in $\{0, 1\}$ (either the drug is effective or it is not). We will consider two examples that illustrate how and when prior information may be useful.

Example 1: Discrete prior Suppose that we know that the mean efficacy of Drug B is $\mu_B = 0.4$, but we have the following prior on Drug A: its mean efficacy is $\mu_A = 0.6$ with probability 0.5, and $\mu_A = 0.2$ with probability 0.5. Concretely, our prior distribution is given by $\boldsymbol{Q}((0.6, 0.4)) = \boldsymbol{Q}((0.2, 0.4)) = 0.5$. Now, suppose that our algorithm has sampled Drug A 3 times, and saw 2 successes and 1 failures. Intuitively, we believe we are more likely witnessing the scenario in which Drug A is superior to Drug B (i.e. the mean effiacy of Drug A is equal to 0.6). We can formally express this belief by evaluating the *posterior* probability that this is the case, via Bayes' rule:

$$\begin{aligned}\boldsymbol{Q}_3(\mu_A = 0.6) &= \frac{\mathbb{P}\left[\text{2/3 successes} \,\middle|\, \mu_A = 0.6\right] \cdot \boldsymbol{Q}((0.6, 0.4))}{\mathbb{P}\left[\text{2/3 successes} \,\middle|\, \mu_A = 0.6\right] \cdot \boldsymbol{Q}((0.6, 0.4)) + \mathbb{P}\left[\text{2/3 successes} \,\middle|\, \mu_A = 0.2\right] \cdot \boldsymbol{Q}((0.6, 0.4))} \\ &\approx 0.81,\end{aligned}$$

and so seeing just 3 samples of Drug A gives us more information to decide which scenario we are in. The approximate calculation at the end substitutes the prior distribution $\boldsymbol{Q}((\cdot, \cdot))$ and calculations of the probabilities of success arising from multiple Bernoulli trials.

Now suppose instead that we had the following prior information on Drug A: its mean efficacy is equal to 0.6 with probability 0.8, and equal to 0.2 with probability 0.2. Concretely, our prior distribution is given by $\boldsymbol{Q}((0.6, 0.4)) = 0.8$ and $\boldsymbol{Q}((0.2, 0.4)) = 0.2$. Then, repeating the above calculation with Bayes' rule would give us $\mathbb{P}\left[\mu_A = 0.6 \middle| \text{2/3 successes}\right] \approx 0.94$. Thus, the prior information has enabled us to go from being 81% sure to 94% sure about the scenario that we are in—in this way, a well-judged prior enables "faster" learning.

Finally, suppose we instead had the following prior information on Drug A: its mean efficiency is equal to 0.6 with probability 0.1, and equal to 0.2 with probability 0.9. Then, repeating the above calculation with Bayes' rule would give us $\mathbb{P}\left[\mu_A = 0.6 \middle| \text{2/3 successes}\right] \approx 0.33$. Thus, the prior information has heavily biased us against the trend of the data—we will need more data to be certain of the outcome! This is an example of a misjudged prior, and clearly it "slows down" learning.

Thus, the examples above highlight two key properties that we would like from a prior distribution: we would like them to be *informative* of the reality, and we would like them to be *structured* for ease of posterior computation from data.

15.1.2 The Bayesian equivalent of "greedy"

Now that we have seen a few examples of prior selection and posterior computation, let us formally recap the Bayesian framework and describe the equivalent of a "greedy" algorithm. Suppose we have begun the learning process with the mean of each arm μ_a being drawn from a prior $\boldsymbol{Q}^{(a)}(\mu_a)$ (note that the priors can be different for each arm, as in both of the examples above where knew exactly the mean μ_B in advance). Further, suppose that have played t rounds and we have pulled arm a $N_a(t)$ times. Then, we can compute posteriors $\boldsymbol{Q}^{(a)}_{N_a(t)}(\mu_a)$ on the means that are functions both of the prior and the observed data. For a given arm index a, we expect that the larger the value of $N_a(t)$, the more the posterior will *contract*, or *concentrate*, towards the true value of the mean μ_a (regardless of what prior is chosen). At this stage, the question that arises is how to utilize these posterior distributions to make a decision. One promising-but-ultimately naive possibility is to use the posteriors to judge the probability that each arm a is the optimal arm, i.e. calculate $q_{t,a} := \boldsymbol{Q}_t\left[\mu_a > \mu_{a'} \text{ for all } a' \neq a\right]$, and select the arm with the maximal value of $q_{t,a}$ at time t. This is often called a *maximum-a-posteriori* (MAP) decision rule. Consider the "discrete prior" Example 1, for which we can verify that $q_{t,A} = \boldsymbol{Q}_t(\mu_A = 0.6)$. The rule of selecting the arm with the maximal value of $q_{t,a}$ would lead us to choose Drug A if $q_{t,a} > 0.5$, and Drug B otherwise.

Thereby, Example 1 illustrates the peril of using this MAP decision rule: its decisions can be too tailored to the prior information, and may not get successfully updated in the way that we want that reflects reality. Suppose, for example, that Nature had given us the true mean of Drug A to be $\mu_A = 0.6$, but we had used a prior belief that $\mu_A = 0.6$ with probability 0.1 and $\mu_A = 0.2$ with probability 0.9. Recall that we also know that the true mean of Drug B is $\mu_B = 0.4$. Then, despite favorable evidence presenting itself at the very beginning (with 2 successful trials and 1 unsuccessful trial), the MAP approach will conclude that Drug A is better only with 33% probability, and will default towards picking the suboptimal Drug B—not only on this round, but on all future rounds. This example

illustrates further the perils of an ill-judged prior, but it also illustrates the perils of choosing a MAP approach, which does not enable us to recover from the initial adverse effects of this ill-judged prior. Just like in the original "greedy" algorithm, this MAP-based approach suffers from the flaw of insufficient exploration. We will now see a "heuristic" way to rescue this that turns out to be very, very successful both in theory and in practice.

15.2. The Thompson sampling algorithm

The key idea in Thompson sampling is to introduce an intrinsic form of *randomization* into the algorithm. In particular, instead of using the posterior to *always* pull the arm which is most likely to be the best, we *sample arm means randomly* from that posterior. Then, we pick the arm that has the maximum "sampled" mean. The randomness in the algorithm arises in the randomness from sampling the posterior, and will become more clear after reading the example provided below the algorithm description.

Algorithm 1 Thompson sampling algorithm

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

The Thompson sampling algorithm is described formally in Algorithm 1. It is most instructive to examine what exactly this algorithm does by starting with the "discrete prior" Example 1. Recall that in Example 1, we know that $\mu_B = 0.4$, and $\mu_A = 0.6$ but we do not know this beforehand. We have seen 3 trials of Drug A, and 2 successes and 1 failure. We now ask how Thompson sampling would make a decision for the three types of priors that we considered:

- In the first choice of "uniform prior", the posterior probability that $\mu_A = 0.6$ was evaluated to be approximately 0.81. Therefore, Thompson sampling will *sample* the posterior as

$$(\mu_{A,t}, \mu_{B,t}) = \begin{cases} (0.6, 0.4) \text{ w.p. } 0.81 \\ (0.2, 0.4) \text{ w.p. } 0.19. \end{cases}$$

  and select the arm with the maximum value of $\mu_{a,t}$. Consequently, Thompson sampling will pull $A_t = A$ with probability 0.81 and $A_t = B$ with probability 0.19. Given that Drug A has been tested only 3 times, this seems like a very reasonable outcome.

- In the second choice of prior that was biased towards $\mu_A = 0.6$, the posterior probability that $\mu_A = 0.6$ was evaluated to be approximately 0.94. In this case, Thompson sampling is more *aggressive*, selecting $A_t = A$ with probability 0.94 even though Drug A has been tested only 3 times. This case illustrates the benefits of Thompson sampling in speeding up learning when we have prior information that Drug A may be more effective than Drug B.

- Lastly, let us consider the third choice of "ill-judged" prior that is biased *against* the truth, i.e. it is only 10% likely to say that $\mu_A = 0.6$. In this case, Thompson sampling will select $A_t = A$ with probability 0.33. While this is a less-than-ideal outcome (we are more likely to select Drug B, even though it is the suboptimal drug), it is still a reasonable one: while the MAP-based approach would have entirely ruled out Drug A, Thompson sampling still lets us sample it 33% of the time. The randomness that arises out of our uncertainty about which option is truly better helps us by allowing us to evaluate Drug A a few more times. Eventually, the Thompson sampling approach can be shown in this example to select Drug A enough times that we eventually recover from the adverse effects of the ill-judged prior.

This case-by-case examination of Thompson sampling illustrates three nice properties about it:

- *It trades off exploration and exploitation naturally by randomizing more when we are less certain about the best outcome.*
- *It speeds up learning by decreasing randomization when the prior is informative and well-judged, i.e. consistent with reality.*
- *It is (somewhat) robust to an ill-judged prior, which can sometimes be the case in practice.*

We will examine these properties in more detail in the first half of next lecture, and conclude this lecture with a brief discussion of implementation considerations.

15.2.1 Implementation considerations

As Thompson sampling takes as input only the prior on means (which is itself an algorithmic design choice), it is a *very* generally applicable algorithm. We will see diverse applications of the Thompson sampling idea throughout the rest of this class. However, whether it can always be implemented *computationally efficiently* is a different matter. In particular, the often most expensive step in Thompson sampling is the step of *computing the posterior distribution*, i.e. Step 3 in Algorithm 1. In the stylized "discrete prior" Example 1, the posterior computations are actually quite complicated and usually not closed-form. We now present a more common example for prior distributions on Bernoulli rewards that is not only more flexible in modeling, but also enables efficient posterior computation.

Example 2: Beta prior A more commonly considered prior distribution for the means of the Bernoulli rewards (of both drugs) is *continuous*, i.e. the means (μ_A, μ_B) are drawn independently from a continuous distribution on the interval $[0, 1]$. A particularly popular choice in practice is the *Beta* distribution on $[0, 1]$, whose probability density function is parameterized by positive-valued $\alpha, \beta > 0$ and defined as $\boldsymbol{Q}_{\mathsf{beta}}(\mu; \alpha, \beta) = \frac{1}{B(\alpha,\beta)} \cdot \mu^{\alpha-1}(1-\mu)^{\beta-1}$. Above, $B(\alpha, \beta)$ is the Beta function and does not depend on μ; consequently, it is simply a normalization constant. Figure 15.1 displays the flexibility in modeling of the Beta distribution for different values of α, β. You may be interested to note that the Beta distribution with the values $\alpha = 1, \beta = 1$ would reduce to the uniform distribution.

Now, let us consider the drug discovery problem, and as before the mean reward of Drug B is known to be $\mu_B = 0.4$, but μ_A is drawn from the prior distribution $\boldsymbol{Q}_{\mathsf{beta}}(\cdot; \alpha, \beta)$. Suppose, as before, that we have seen 3 samples from Drug A with 2 successes and 1 failure. Then, we are interested in using these samples to compute a *posterior* distribution on what we believe the mean reward of Drug A is that combines the prior and our observations. It turns out that this distribution, which we denote by $\boldsymbol{Q}_3(\mu)$, is *itself* a Beta distribution, only with adjusted parameters $(\alpha + 2, \beta + 1)$! In other words, we have $\boldsymbol{Q}_3(\mu) = \boldsymbol{Q}_{\mathsf{beta}}(\mu; \alpha + 2, \beta + 1)$. More generally, if we conducted n trials of Drug A and observed k successes, the posterior would be given by $\boldsymbol{Q}_n(\mu) = \boldsymbol{Q}_{\mathsf{beta}}(\mu; \alpha + k, \beta + n - k)$. For example, if there are more successes than failures, we should expect the peak of $\boldsymbol{Q}_n(\mu)$ to move towards 1 (as depicted in Figure 15.1a for our example of 3 trials, 2 successes and 1 failure, and the prior choice $\alpha = \beta = 1$). The Beta prior example illustrates the benefits of a specially structured prior: in this case, a posterior distribution can be very easily computed (unlike in the discrete-prior case), and is in fact from the same Beta family as the prior distribution.

This is an example of what is commonly called a *conjugate prior* (see the review lecture on probability for further details on conjugate priors and what they mean). It leads to the greatly simplified implementation of Thompson sampling, described below.

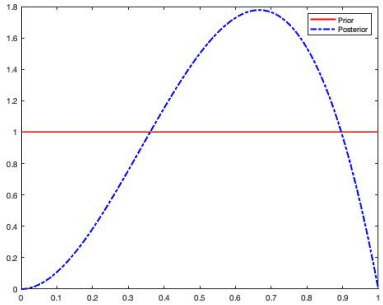
Algorithm 2 Thompson sampling algorithm for Bernoulli rewards, Beta prior

Input: Beta prior parameters (α_a, β_a) on arm a for $a = 1, \ldots, K$
for $t = 1, \ldots, T$ **do**
  Compute Beta posterior distribution parameters $(\alpha_a + K_a(t), \beta_a + N_a(t) - K_a(t))$ on μ_a from observed samples
  Sample $(\mu_{1,t}, \mu_{2,t}, \ldots, \mu_{K,t})$ from the posterior distributions given by parameters above
  Pull arm $A_t = \arg\max_{a \in \{1,\ldots,K\}} \mu_{a,t}$, and observe reward G_{t,A_t}.
  Update total number of pulls $N_{A_t,t} = N_{A_t,t} + 1$, number of successes $K_{A_t,t} = K_{A_t,t} + G_{t,A_t}$.
end for

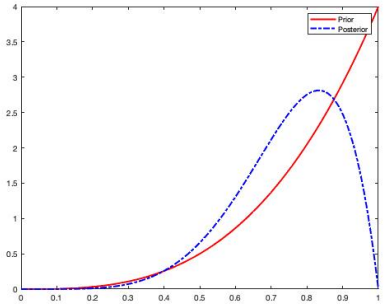
While Example 2 does not admit as clean a case-by-case evaluation of Thompson sampling as Example 1, the qualitative ideas turn out to be similar, and we will still have three representative cases for evaluation, listed below:

- If the initial parameters $\alpha_a = \beta_a$, then we are roughly in a "uniform prior" situation.
- If $\alpha_a \gg \beta_a$, then we believe that the mean of arm a is more likely to be closer to 1 than 0, and the posterior moves very speedily towards this case (as illustrated in Figure 15.1b).
- If $\alpha_a \ll \beta_a$, then we believe that the mean of arm a is more likely to be closer to 0 than 1 (as illustrated in Figure 15.1c), and if arm a turns out to be optimal, we need several more samples to figure this out. Regardless, Figure 15.1c illustrates that the posterior moves in the right direction, and sampling from it still allows the desired action $A_t = A$ with at least some non-zero probability.

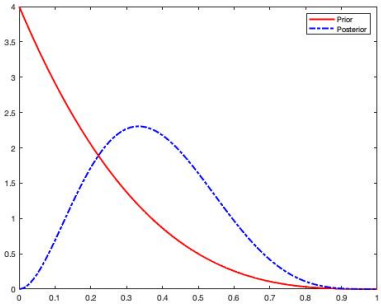
For the first half of next lecture, we will briefly discuss guarantees on Thompson sampling, and specially focus on the guarantees which highlight its potential benefits over UCB.



(a) Uniform prior $\alpha = \beta = 1$



(b) Well-specified prior $\alpha = 4, \beta = 1$ (more likely to judge $\mu_A > \mu_B$ than not).



(c) Misspecified prior $\alpha = 1, \beta = 4$ (less likely to judge $\mu_A > \mu_B$ than not).

Figure 15.1: Evolution of the Beta prior and posterior for various values of α, β after 2 successes and 1 failure have been observed. The prior distributions are marked in red, and the posterior distributions are marked in blue.

15.3. Additional references

- See the recently authored tutorial on Thompson sampling (Russo et al., 2018) for an excellent introduction to Thompson sampling in its own right. To better understand the basics of Thompson sampling, I particularly recommend Chapter 3, which examines Thompson sampling in depth for the Bernoulli bandit and Beta prior example, and Chapter 4 which examines Thompson sampling in a general framework. If you are interested in implementation and modeling considerations, I also recommend Chapters 5 and 6. Chapter 7 touches upon applications of Thompson sampling in more complex domains like contextual bandits and reinforcement learning—time permitting, we will return to this when we discuss reinforcement learning in November.

- Thompson sampling was introduced in 1933 in the seminal paper (Thompson, 1933), but its theoretical properties were poorly understood until the last decade! Papers like Chapelle and Li (2011) demonstrated the empirical benefits of Thompson sampling over UCB in practice (perhaps owing to the flexibility that we have described of incorporating informative priors) and led to a resurgence of interest in understanding its properties, some of which we will touch upon next lecture.

- We have provided a qualitative argument here to justify why Thompson sampling will (eventually) recover from an ill-judged prior. However, the exact quantitative impact of such misspecification on the performance of Thompson sampling is a nuanced issue, and continues to be researched till today.

References

Olivier Chapelle and Lihong Li. An empirical evaluation of thompson sampling. In *Advances in Neural Information Processing Systems 24*, pages 2249–2257, 2011.

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.

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