

ECE 8803: Online Decision Making in Machine Learning

Homework 4

Released: Oct 28

Due: Nov 10

Objective. To facilitate theoretical and empirical exploration of the Thompson sampling and linear UCB algorithms that were introduced in the last 2 weeks of class, with the aim of deepening understanding about these bandit algorithms.

Problem 1 (Is Thompson sampling always optimal?) 20 points Thompson sampling and UCB are two of the most popular algorithms for the multi-armed bandit problem. We have also seen evidence for their *optimality*, but only under the assumption that the arms' reward mean parameters are independently chosen: for example, pulling arm 1 does not yield any partial information about the other arms. In this problem, we will examine what happens if arms do reveal partial information about each other: is Thompson sampling still the best idea?

In particular, we will consider a 4-armed bandit instance, and θ to be an unknown parameter that is drawn uniformly at random from the set $\{1, 2, 3\}$. In our model, the parameter θ will influence all of the rewards of the 4 arms. Then, the first 3 arms will yield Bernoulli rewards (i.e. in $\{0, 1\}$) with means given by:

$$(\mu_1, \mu_2, \mu_3) = \begin{cases} (\frac{2}{3}, \frac{1}{2}, \frac{1}{3}) & \text{if } \theta = 1 \\ (\frac{1}{3}, \frac{2}{3}, \frac{1}{2}) & \text{if } \theta = 2 \\ (\frac{1}{2}, \frac{1}{3}, \frac{2}{3}) & \text{if } \theta = 3. \end{cases}$$

Moreover, the fourth arm will yield *deterministic* reward equal to $\frac{1}{4\theta}$, i.e. there is no noise in the reward observation. While our algorithms will not know the value of θ , they will know that the reward of the 4-th arm is deterministic.

In this problem, we will evaluate: a) the pseudo-regret for each value of θ , and b) the Bayesian regret over the uniform distribution that is specified on θ .

- (a) Report the identity of the optimal arm in each of the cases $\theta = 1, 2, 3$.
- (b) Consider the Thompson sampling algorithm run with the uniform prior over the instances $\theta = 1, 2, 3$. Evaluate the probability distribution over the action taken at the first round, which is denoted by A_1 .
- (c) Suppose that Nature gave us $\theta = 1$ (therefore, the true instance is given by $\mu_1 = 2/3, \mu_2 = 1/2, \mu_3 = 1/3, \mu_4 = 1/4$). Evaluate the probability distribution over the action taken at the second round, which is denoted by A_2 , in all six cases: a) $A_1 = 1$ and observed reward equal to 1, b) $A_1 = 1$ and observed reward equal to 0, c) $A_1 = 2$ and observed reward equal to 1, d) $A_1 = 2$ and observed reward equal to 0, e) $A_1 = 3$ and observed reward equal to 1, f) $A_1 = 3$ and observed reward equal to 0.

- (d) Based on the answers to parts (b) and (c), is there an arm that will never be pulled by Thompson sampling *on any round*? Justify your answer either way.

Hint: use Bayes' rule for posterior computation at round t to justify your answer formally.

- (e) Design an algorithm that will pull *only one* of the suboptimal arms at most *once*, thereby achieving a pseudo-regret of *at most* $\frac{2}{3}$. Show that the same guarantee is achieved by this algorithm for the cases $\theta = 2$ and $\theta = 3$ as well.

Hint: what is the sufficient piece of information needed to figure out the identity of the optimal arm, and how can you get it most quickly?

- (f) Evaluate the Bayesian regret of the algorithm that you designed, and compare it with the Bayesian regret guarantee that we can obtain for Thompson sampling.

Problem 2 (Writing LinUCB more explicitly) 10 points For a positive semidefinite matrix \mathbf{M} , we write the weighted norm of a vector \mathbf{x} as $\|\mathbf{x}\|_{\mathbf{M}} := \sqrt{\mathbf{x}^\top \mathbf{M} \mathbf{x}}$. We also define the regression Gram matrix $\mathbf{V}_t := \lambda \mathbf{I} + \sum_{s=1}^t \mathbf{A}_s \mathbf{A}_s^\top$ for all values of t , where λ is a hyperparameter.

In Lecture 17, we saw that the decision that is taken by the LinUCB algorithm at round t can be written as

$$\mathbf{A}_t = \arg \max_{\mathbf{a} \in \mathcal{A}} \max_{\boldsymbol{\theta} \in \mathcal{C}_t} \langle \mathbf{a}, \boldsymbol{\theta} \rangle, \quad (1)$$

where we defined

$$\mathcal{C}_t := \{\boldsymbol{\theta} : \|\boldsymbol{\theta} - \widehat{\boldsymbol{\theta}}_t\|_{\mathbf{V}_{t-1}}^2 \leq \beta_t\}$$

and β_t is a hyperparameter for round t .

In this problem, you will show through a sequence of steps that you can rewrite Equation (1) in a more explicit and interpretable form that trades off exploration and exploitation. You will use this explicit form in Problem 3 of the HW to implement LinUCB.

- (a) For any value of $\boldsymbol{\theta} \in \mathcal{C}_t$, denote $\mathbf{e} := \boldsymbol{\theta} - \widehat{\boldsymbol{\theta}}_t$. Use the definition of the confidence set \mathcal{C}_t to show that Equation (1) can be rewritten as

$$\mathbf{A}_t = \arg \max_{\mathbf{a} \in \mathcal{A}} \left[\langle \mathbf{a}, \widehat{\boldsymbol{\theta}}_t \rangle + \max_{\|\mathbf{e}\|_{\mathbf{V}_{t-1}}^2 \leq \beta_t} \langle \mathbf{a}, \mathbf{e} \rangle \right] \quad (2)$$

- (b) Next, show that Equation (2) can be rewritten as

$$\mathbf{A}_t = \arg \max_{\mathbf{a} \in \mathcal{A}} \left[\langle \mathbf{a}, \widehat{\boldsymbol{\theta}}_t \rangle + \max_{\|\mathbf{f}\|_2^2 \leq \beta_t} \langle \mathbf{V}_{t-1}^{-1/2} \mathbf{a}, \mathbf{f} \rangle \right],$$

where we define $\mathbf{f} := \mathbf{V}_{t-1}^{1/2} \mathbf{e}$.

- (c) Use the Cauchy-Schwarz inequality to show that

$$\max_{\|\mathbf{f}\|_2^2 \leq \beta_t} \langle \mathbf{V}_{t-1}^{-1/2} \mathbf{a}, \mathbf{f} \rangle = \sqrt{\beta_t} \cdot \|\mathbf{a}\|_{\mathbf{V}_{t-1}^{-1}},$$

and so we can ultimately rewrite the LinUCB update as

$$\mathbf{A}_t = \arg \max_{\mathbf{a} \in \mathcal{A}} \left[\langle \mathbf{a}, \widehat{\boldsymbol{\theta}}_t \rangle + \sqrt{\beta_t} \cdot \|\mathbf{a}\|_{\mathbf{V}_{t-1}^{-1}} \right] \quad (3)$$

Hint: Use the fact that you can write $\mathbf{x}^\top \mathbf{M} \mathbf{x} = \|\mathbf{M}^{1/2} \mathbf{x}\|_2^2$.

- (d) (BONUS — 5 points) Justify how the optimization problem in Equation (3) trades off exploration and exploitation by answering the following questions in words:

- Which term incentivizes exploration and why?
- Which term incentivizes exploitation and why?

Problem 3 (Empirical comparison of LinUCB and UCB over finite action sets) 20 points In this problem, you will implement and compare LinUCB and UCB on 2 different cases: one where the action set is very small, and the other where the action set is very large (in both cases, the action set will be discrete). This problem is a reproduction of Problem 19.8 in the book “Bandit Algorithms” by Tor Lattimore and Csaba Szepesvari. You will find it useful to refer to the figures provided in the corresponding problem in the online copy: <https://tor-lattimore.com/downloads/book/book.pdf>, and you will also find it useful to refer to Equation (3) in Problem 2 to implement LinUCB. In particular, you will implement LinUCB by evaluating Equation (3) on all actions in the discrete action set \mathcal{A} and computing the minimum via brute force. If the minimizing action is not unique, sample among all minimizing actions uniformly at random.

For LinUCB, you will use hyperparameters $\beta_t = \beta := 1 + \sqrt{2 \log T + d \log \left(\frac{d+T}{d}\right)}$ and $\lambda = 1$, and for UCB, you will use the hyperparameter $\delta = 1/T^2$. Please provide your solutions to this problem in a Jupyter notebook.

- (a) (5 points) First, implement LinUCB and UCB on a 2-armed bandit instance for $T = 1000$ rounds. The reward distributions of both the arms are Gaussian with unit variance and means given by $(\mu_1 = 0, \mu_2 = -\Delta)$ where $\Delta \in [0, 1]$. For LinUCB, you will choose the action set $\mathcal{A} = \{\mathbf{e}_1, \mathbf{e}_2\}$ comprised of the 2-dimensional standard basis vectors.
- (b) (5 points) Plot the pseudoregret of LinUCB and UCB as a function of Δ by averaging the quantity by averaging the quantity $\sum_{a \neq a^*} \Delta_a \cdot N_T(a)$ over 100 runs for each value of Δ . Your eventual figure should resemble the left plot in Figure 19.1 in <https://tor-lattimore.com/downloads/book/book.pdf>.
- (c) (5 points) Implement LinUCB and UCB on a k -armed linear bandit instance where $2 \leq k \leq 1000$, with the following specifications:
 - Fix $d = 5$, and the unknown parameter is given by $\boldsymbol{\theta}^* = \frac{1}{\sqrt{5}} \mathbf{1}$, where $\mathbf{1}$ denotes the all 1’s vector.
 - Draw 1000 vectors $\{\mathbf{a}_1, \dots, \mathbf{a}_{1000}\}$ uniformly at random from the unit sphere, i.e. $\|\mathbf{a}_i\|_2 = 1$ for all $i = 1, \dots, k$. Then, for each choice of k , we consider the action set $\mathcal{A}_k = \{\mathbf{a}_1, \dots, \mathbf{a}_k\}$ for LinUCB. Correspondingly, the mean of arm a will be $\mu_a := \langle \mathbf{a}_a, \boldsymbol{\theta}^* \rangle$.
 - The noise W_t is standard Gaussian at each round t .
- (d) Plot the pseudoregret of LinUCB and UCB as a function of k by averaging the quantity $\sum_{a \neq a^*} \Delta_a \cdot N_T(a)$ over 100 runs for each value of k . Your eventual figure should resemble the right plot in Figure 19.1 in <https://tor-lattimore.com/downloads/book/book.pdf>.
- (e) (BONUS — 10 points) Justify through words why LinUCB is the superior algorithm in part (d). Justify through a mathematical explanation why UCB is the superior algorithm in part (b).

Hint: for the case that is evaluated in parts (a) and (b), write the LinUCB algorithm as an instance of UCB with a different choice of hyperparameter.