Fall 2022 *Instructor: Jacob Abernethy, Bhuvesh Kumar, Guanghui Wang*

ECE 8803: Online Decision Making in Machine Learning

Homework 4

Released: Nov 27 *Due: Dec 15*

Problem 1 (Writing LinUCB more explicitly) 15 points For a positive semidefinite matrix $\boldsymbol{M}$, we write the weighted norm of a vector $\boldsymbol{x}$ as $\|\boldsymbol{x}\|_{\boldsymbol{M}} := \sqrt{\boldsymbol{x}^\top \boldsymbol{M}\boldsymbol{x}}$. We also define the regression Gram matrix $\boldsymbol{V}_t := \lambda \boldsymbol{I} + \sum_{s=1}^{t} \boldsymbol{A}_s \boldsymbol{A}_s^\top$ for all values of t, where λ is a hyperparameter.

In Lecture 21, we saw that the decision that is taken by the LinUCB algorithm at round t can be written as

$$\boldsymbol{A}_t = \arg\max_{\boldsymbol{a}\in\mathcal{A}} \max_{\boldsymbol{\theta}\in\mathcal{C}_t} \langle \boldsymbol{a}, \boldsymbol{\theta}\rangle, \tag{1}$$

where we defined

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

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

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

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

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

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

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

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

and so we can ultimately rewrite the LinUCB update as

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

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

(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 2 (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 $\mathcal{A} = \{\boldsymbol{e}_1, \boldsymbol{e}_2\}$ 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 $\{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_{1000}\}$ uniformly at random from the unit sphere, i.e. $\|\boldsymbol{a}_i\|_2 = 1$ for all $i = 1, \ldots, k$. Then, for each choice of k, we consider the action set $\mathcal{A}_k = \{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_k\}$ for LinUCB. Correspondingly, the mean of arm a will be $\mu_a := \langle \boldsymbol{a}_a, \boldsymbol{\theta}^* \rangle$.
- 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.