CS 234 Winter 2021
HW 4
Due: March 17 at 6:00 pm (PST)

For submission instructions please refer to website For all problems, if you use an existing result from either the literature or a textbook to solve the exercise, you need to cite the source.

1 Estimation of the Warfarin Dose [60 pts]

1.1 Introduction

Warfarin is the most widely used oral blood anticoagulant agent worldwide; with more than 30 million prescriptions for this drug in the United States in 2004. The appropriate dose of warfarin is difficult to establish because it can vary substantially among patients, and the consequences of taking an incorrect dose can be severe. If a patient receives a dosage that is too high, they may experience excessive anti-coagulation (which can lead to dangerous bleeding), and if a patient receives a dosage which is too low, they may experience inadequate anti-coagulation (which can mean that it is not helping to prevent blood clots). Because incorrect doses contribute to a high rate of adverse effects, there is interest in developing improved strategies for determining the appropriate dose (Consortium, 2009).

Commonly used approaches to prescribe the initial warfarin dosage are the *pharmacogenetic algorithm* developed by the IWPC (International Warfarin Pharmacogenetics Consortium), the *clinical algorithm* and a *fixed-dose* approach.

In practice a patient is typically prescribed an initial dose, the doctor then monitors how the patient responds to the dosage, and then adjusts the patient's dosage. This interaction can proceed for several rounds before the best dosage is identified. However, it is best if the correct dosage can be initially prescribed.

This question is motivated by the challenge of Warfarin dosing, and considers a simplification of this important problem, using real data. The goal of this question is to explore the performance of multi-armed bandit algorithms to best predict the correct dosage of Warfarin for a patient *without* a trial-an-error procedure as typically employed.

Problem setting Let T be the number of time steps. At each time step t, a new patient arrives and we observe its individual feature vector $X_t \in \mathbb{R}^d$: this represents the available knowledge about the patient (e.g., gender, age, ...). The decision-maker (your algorithm) has access to K arms, where the arm represents the warfarin dosage to provide to the patient. For simplicity, we discretize the actions into $K = 3$

- Low warfarin dose: under 21mg/week
- Medium warfarin dose: 21-49 mg/week
- High warfarin dose: above 49mg/week

If the algorithm identifies the correct dosage for the patient, the reward is 0, otherwise a reward of -1 is received.
Lattimore and Szepesvári have a nice series of blog posts that provide a good introduction to bandit algorithms, available here: BanditAlgs.com. The Introduction and the Linear Bandit posts may be particularly of interest. For more details of the available Bandit literature you can check out the Bandit Algorithms Book by the same authors.

1.2 Dataset

We use a publicly available patient dataset that was collected by staff at the Pharmacogenetics and Pharmacogenomics Knowledge Base (PharmGKB) for 5700 patients who were treated with warfarin from 21 research groups spanning 9 countries and 4 continents. You can find the data in `warfarin.csv` and metadata containing a description of each column in `metadata.xls`. Features of each patient in this dataset includes, demographics (gender, race, . . .), background (height, weight, medical history, . . .), phenotypes and genotypes.
Importantly, this data contains the true patient-specific optimal warfarin doses (which are initially unknown but are eventually found through the physician-guided dose adjustment process over the course of a few weeks) for 5528 patients. You may find this data in mg/week in `Therapeutic Dose of Warfarin`[1] column in `warfarin.csv`. There are in total 5528 patient with known therapeutic dose of warfarin in the dataset (you may drop and ignore the remaining 173 patients for the purpose of this question). Given this data you can classify the right dosage for each patient as *low*: less than 21 mg/week, *medium*: 21-49 mg/week and *high*: more than 49 mg/week, as defined in Consortium (2009) and Introduction.
The data processing is already implemented for you

1.3 Implementing Baselines [10 pts]

Please implement the following two baselines in main.py

1. *Fixed-dose*: This approach will assign 35mg/week (medium) dose to all patients.
2. *Warfarin Clinical Dosing Algorithm*: This method is a linear model based on age, height, weight, race and medications that patient is taking. You can find the exact model is section S1f of `appx.pdf`.

Run the fixed dosing algorithm and clinical dosing algorithm with the following command:

```
python main.py --run-fixed --run-clinical
```

You should see the total_fraction_correct to be fixed at about 0.61 for fixed dose and 0.64 for clinical dose algorithm. You can run them individually as well. Just use one of the command line arguments instead.

[1] You cannot use `Therapeutic Dose of Warfarin` data as an input to your algorithm.

1.4 Implementing a Linear Upper Confidence Bandit Algorithm [15 pts]

Please implement the Disjoint Linear Upper Confidence Bound (LinUCB) algorithm from Li et al. (2010) in main.py. See Algorithm 1 from paper. Please feel free to adjust the –alpha argument, but you don't have to. Run the LinUCB algorithm with the following command:

```
python main.py --run-linucb
```

You should see the total_fraction_correct to be above 0.64, though the results may vary per run.

1.5 Implementing a Linear eGreedy Bandit Algorithm [5 pts]

Is the upper confidence bound making a difference? Please implement the e-Greedy algorithm in main.py. Please feel free to adjust the –ep argument, but you don't have to. Does eGreedy perform better or worse than Upper Confidence bound? (You do not need to include your answers here) Run the ε-greedy LinUCB with the following command:

```
python main.py --run-egreedy
```

You should see the total_fraction_correct to be above 0.61, though the results may vary per run.

1.6 Implementing a Thompson Sampling Algorithm [20 pts]

Please implement the Thompson Sampling for Contextual Bandits from Agrawal and Goyal (2013) in main.py. See Algorithm 1 and section 2.2 from paper. Please feel free to adjust the –v2 argument, but you don't have to. (This actually v squared from the paper) Run the Thompson Sampling algorithm with the following command:

```
python main.py --run-thompson
```

You should see the total_fraction_correct to be **around** 0.64, though the results may vary per run.

1.7 Results [10 pts]

At this point, you should see a plot in your results folder titled "fraction_incorrect.png". If not, run the following command to generate the plot:

```
python main.py
```

Include this plot in for this part. Please also comment on your results in a few sentences. How would you compare the algorithms? Which algorithm "did the best" based on your metric?

2 A Bayesian regret bound for Thompson sampling [40 pts]

Consider the K-armed bandit problem: there are K "arms" (actions), and we will choose one arm $a_t \in [K]$ to pull at each time $t \in [T]$, then receive a random reward $r_t \sim p(r \mid \theta, a = a_t)$. Here θ is a random variable that parameterizes the reward distribution. Its "true" value is unknown to us, but we can make probabilistic inferences about it by combining prior belief with observed reward data. We denote the expected reward for arm a (for a fixed θ) as $\mu_\theta(a) := \mathbb{E}[r \mid \theta, a]$.

A *policy* specifies a distribution over the next arm to pull, given the observed history of interactions $H_t = (a_1, r_1, \ldots, a_{t-1}, r_{t-1})$.[2] Formally, a policy is a collection of maps $\pi = \{\pi_t : \mathcal{H}_t \to \Delta(\mathcal{A})\}_{t=1}^T$, where $\mathcal{H}_t$ is the space of all possible histories at time t and $\Delta(\mathcal{A})$ is the set of probability distributions over $\mathcal{A}$. We denote the probability of arm a under policy π at time t as $\pi_t(a \,|\, H_t)$.

For a fixed value of θ, the suboptimality of a policy π can be measured by the *expected regret*:

$$R_{T,\theta}(\pi) = \mathbb{E}_H \left[\sum_{t=1}^T \mu_\theta(a^*) - \mu_\theta(a_t) \,\middle|\, \theta \right]$$

where the expectation is taken with respect to the arms selected, $a_t \sim \pi_t(a \,|\, H_t)$, and rewards subsequently observed, $r_t \sim p(r \,|\, \theta, a = a_t)$. We use H as a shorthand for $H_{T+1} = (a_1, r_1, \ldots, a_T, r_T)$.[3] Note that a^* is random because θ is random, but for a given θ it is fixed and can be computed by $a^* = \arg\max_a \mu_\theta(a)$. (Assume for simplicity that there is one optimal action for any given θ.)

Our goal in this problem is to prove a bound on the *Bayesian regret*, which is the expected regret averaged over a prior distribution on θ:

$$\mathrm{BR}_T(\pi) = \mathbb{E}_\theta[R_{T,\theta}(\pi)]$$

We will analyze the *Thompson sampling* (or *posterior sampling*) algorithm, which operates by sampling from the posterior distribution of the optimal action a^* given H_t:

$$\pi_t^{\mathrm{TS}}(a \,|\, H_t) = p(a^* = a \,|\, H_t)$$

We can sample from π_t^{TS} by first sampling $\theta_t \sim p(\theta \,|\, H_t)$ and then computing $a_t = \arg\max_a \mu_{\theta_t}(a)$.

(a) [7 pts] Let $\{L_t : \mathcal{A} \to \mathbb{R}\}_{t=1}^T$ and $\{U_t : \mathcal{A} \to \mathbb{R}\}_{t=1}^T$ be lower and upper confidence bound[4] sequences (respectively), where each L_t and U_t depends on H_t. Show that the Bayesian regret for Thompson sampling can be decomposed as

$$\mathrm{BR}_T(\pi^{\mathrm{TS}}) = \mathbb{E}_{\theta,H}\left[\sum_{t=1}^T [U_t(a_t) - L_t(a_t)] + [L_t(a_t) - \mu_\theta(a_t)] + [\mu_\theta(a^*) - U_t(a^*)] \right]$$

This equality does not hold in general, so its proof will require using some property of π^{TS}.

(b) [8 pts] Now assume the rewards r_t are bounded in $[0, 1]$ and $L_t \leq U_t$. Show that

$$\mathrm{BR}_T(\pi^{\mathrm{TS}}) \leq \mathbb{E}_{\theta,H}\left[\left(\sum_{t=1}^T [U_t(a_t) - L_t(a_t)] \right) + T \sum_a \mathbb{I}\left\{ \bigcup_{t=1}^T \{\mu_\theta(a) \notin [L_t(a), U_t(a)]\} \right\} \right]$$

where the notation $\mathbb{I}\{\cdot\}$ refers to an indicator random variable which equals 1 if the expression inside the brackets is true and equals 0 otherwise.

[2] Note: we take history to mean that which is known at the beginning of step t, rather than at the end of step t, so it only goes up to a_{t-1}, r_{t-1}.

[3] The regret does not actually depend on r_T.

[4] In Thompson sampling, the upper confidence bound is not used to select actions; we only introduce it for the purpose of analysis.

Let us now impose a specific form of confidence bounds:

$$L_t(a) = \max\left\{0, \hat{\mu}_t(a) - \sqrt{\frac{2 + 6\log T}{n_t(a)}}\right\}$$
$$U_t(a) = \min\left\{1, \hat{\mu}_t(a) + \sqrt{\frac{2 + 6\log T}{n_t(a)}}\right\}$$

where $\hat{\mu}_t(a)$ is the mean of rewards received playing action a before time t, and $n_t(a)$ is the number of times action a was played before time t. (If action a has never been played at time t, $L_t(a) = 0$ and $U_t(a) = 1$.)

You may take as given the following fact: with L_t and U_t defined as above, it holds that

$$\forall a, \quad \mathbb{P}_{\theta,H}\left(\bigcup_{t=1}^{T}\{\mu_\theta \notin [L_t(a), U_t(a)]\}\right) \leq \frac{1}{T}$$

and thus, the bound from part (b) implies

$$\mathrm{BR}_T(\pi^{\mathrm{TS}}) \leq \mathbb{E}_{\theta,H}\left[\sum_{t=1}^{T}[U_t(a_t) - L_t(a_t)]\right] + K$$

To bound the remaining terms, let us use the decomposition $\sum_{t=1}^{T}[U_t(a_t) - L_t(a_t)] = \sum_a \sum_{t\in\mathcal{T}_a}[U_t(a) - L_t(a)]$, where $\mathcal{T}_a = \{t \in [T] : a_t = a\}$.

(c) [5 pts] Show that

$$\sum_{t\in\mathcal{T}_a}[U_t(a) - L_t(a)] \leq 1 + 2\sqrt{2 + 6\log T}\sum_{i=1}^{n_T(a)}\frac{1}{\sqrt{i}}$$

(d) [7 pts] Show that

$$\sum_{i=1}^{n_T(a)}\frac{1}{\sqrt{i}} \leq 2\sqrt{n_T(a)}$$

(Hint: Bound the sum by an integral.)

(e) [8 pts] Use the previous parts to obtain

$$\mathrm{BR}_T(\pi^{\mathrm{TS}}) \leq 2K + 4\sqrt{KT(2 + 6\log T)}$$

(Hint: You may find the *AM-QM inequality* $\frac{1}{n}\sum_{i=1}^{n} x_i \leq \sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}$ helpful.)

(f) [5 pts] Suppose the prior over θ is wildly misspecified, such that the prior probability of the true θ is extremely small or zero. What goes wrong in the regret analysis we have done above?

References

S. Agrawal and N. Goyal. Thompson sampling for contextual bandits with linear payoffs. In *International Conference on Machine Learning*, pages 127–135, 2013.

I. W. P. Consortium. Estimation of the warfarin dose with clinical and pharmacogenetic data. *New England Journal of Medicine*, 360(8):753–764, 2009.

L. Li, W. Chu, J. Langford, and R. E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670, 2010.