CS 234 (Winter 2023)
Default Final Project
Estimation of the Warfarin Dose

Due date: Wednesday March 22 at 6:00 pm PST.
Late days: You cannot use late day on the final project report and poster submission. Please read the late day policy on website for more details.

This project can be done in groups up to 3. All group members should have full understanding of the submitted work. You may use any existing code, libraries, etc. and consult any papers, books, online references, etc. for this project. However, you must cite your sources in your writeup and clearly indicate which parts of the project are your contributions and which parts were implemented by others. Under no circumstances may you look at another group's code or incorporate their code into your project.
Read the section on Academic Collaboration and Misconduct for an overview of the collaboration policy and academic integrity standards expected in general.

Contents

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 project is motivated by the challenge of Warfarin dosing, and considers a simplification of this important problem, using real data. The goal of this project 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.

We assume that the reward for each arm depends on the patient features. For simplicity, consider a linear model where the reward for arm $i \in [K]$ for a patient with features X_t is

$$r_t(X_t, a_i) = X_t^T \beta_i + \varepsilon_{i,t}$$

where $\beta_i \in \mathbb{R}^d$ is an unknown parameter and and the $\varepsilon_{i,t}$ are independent Gaussian random variables with zero mean.

The goal of this project is to design a bandit algorithm that learns a mapping $\pi : X \to a$ that yields the maximal expected reward. Let $\pi_t(s) \in [K]$ denote the arm chosen by policy π at time $t \in [T]$ for the patient s_t. Define the optimal policy $\pi^\star$ to be the policy that maximizes the expected reward across patients given the true β_j parameters for $j = 1, \cdots, K$: this algorithm always chose the the

arm with $\max_j(X_t^T\beta_j)$. Under the linear model, if the agent chooses arm i at timestep t it will incur the expected regret of

$$\mathbb{E}[\max_j[X_t^T\beta_j] - X_t^T\beta_i]$$

The goal is to create and evaluate algorithms that minimize the cumulative expected regret $R_T = \sum_{t=1}^{T} \mathbb{E}[\max_j[X_t^T\beta_j] - X_t^T\beta_i]$.

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.

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 project). 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.

Missing Data: There are missing values in the dataset, you may impute some of the missing values, or just treat "missing" or "unknown" as an extra possible feature value: for example, gender can be {male, female, unknown}. In `appx.pdf` section S4 you may find guidelines on how to impute some of the missing genotypes.

Baselines [10pts]

To become familiar with the data the first task is to measure the performance of existing approaches. As described in Introduction there exist three main approaches for determining the initial warfarin dose

1. *Fixed-dose*: This approach will assign 35mg/day (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`.

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

3. *Warfarin Pharmacogenetic Dosing Algorithm*: This method is also a linear model proposed by Consortium (2009), and it also includes genotypes as an input. You can find the exact model is section S1e of `appx.pdf`.

Choose two of the three models above, and report the performance of these methods on the dataset. **Performance:** the performance metric is the fraction of incorrect decisions an algorithm makes. We say an algorithm has made an incorrect decision if the action is not the same as bucketed dosage in column `Therapeutic Dose of Warfarin` of the dataset.

Implementing a Linear Bandit Algorithm on Warfarin Dataset [50pts]

Your task is to implement a bandit algorithm that outperforms at least the *fixed-dose* baseline and hopefully the other one. Your bandit algorithm will start with just the knowledge of the action set and the features map (you can choose any subset of features from the dataset, you may look at section S1a-S1d in `appx.pdf` for relevant features used in the baseline algorithms. You cannot use the ground truth, `Therapeutic Dose of Warfarin`, as a feature). When a patient comes in, your bandit algorithm makes a recommendation on the amount of Warfarin for that patient (low, medium, high) and observes either a reward of 0 or -1, corresponding to a correct, incorrect decision respectively. your algorithm will then use the observed reward to update its parameters.
To simulate an online learning environment you will need to sample (without replacement) patients from the provided dataset and pass their features to the algorithm. The regret is evaluated while the algorithm learns on the patients provided to the algorithm: you can assume that the optimal policy always makes the right decision.

Algorithm: One option to solve this question is to use a linear bandit algorithm. An overview of these algorithms is given here, though which variant to use is up to you. The references at the end of this document will give you potential directions.
Performance: You should report two measures of performance. *Regret* as defined in section Introduction, and *the fraction of incorrect dosing decisions*, the fraction of incorrect dosing decisions made so far by the algorithm at the current timestep out of all patients seen so far by the algorithm at the current timestep. Plot the performance of your bandit algorithm along with the performance of the baselines you implemented in the section above on these two performance measures.
Robustness: In order to make sure your performance gain/ loss over baselines is not because of ordering of the patients, run your algorithms multiple times on different random permutations of the patients. Do at least 20 runs on random permutations of patients and plot the performance with 95% confidence intervals (use the T-Distribution). Make sure to do this for both the specified measures of performance.

Further Directions [40pts]

There are multiple avenues you can follow. We propose a few suggestions here, but you are free to chose the direction of your project from here. Note that some of our suggested extensions are harder than others and you may come up with your own extension that may be harder than some listed here. We will take into consideration that some directions are harder than others during grading.

- **Reward Structure and Risk Sensitivity**: The reward is currently defined as discrete values $\{-1, 0\}$. An interesting issue is to explore using real-valued rewards and study the performance and regret of your algorithm. The exact reward function is up to you, but it should intuitively make sense. For example, you could create a risk-sensitive reward that accounts for the fact that giving a very high dose to a patient that requires only a low dose can be more harmful that the converse. You could also measure the performance of your algorithm in terms risk-sensitive metrics such as the number of severe mistakes it makes.

- **Other Algorithms** You can compare your algorithm to other Bandits algorithms, for example the Lasso Bandit Bastani and Bayati (2015), which uses a special type of regularization to improve performance. Bastani and Bayati (2015) used the same dataset and problem setup as you did. Other possible algorithms (unless you used them in the previous part) include the LinUCB algorithm Li et al. (2010), Thompson Sampling for Contextual Bandits Agrawal and Goyal (2013), or robust algorithms Neu and Olkhovskaya (2020).

- **Fairness** Fairness recently has been a popular topic in Machine Learning. You can explore fairness by implementing training algorithms for contextual bandits that provide fairness guarantees such as Robinhood Metevier et al. (2019) to try to learn an algorithm that is fair across subgroups. One idea in this direction is to try partitioning the data into a train and test set and use the test set to compare the performance of your original algorithm from the part above (retrained using only your new train partition) and your fairness aware algorithm for the different subgroups.

- **Non-Linear Models**: Thompson sampling usually requires computing a posterior distribution exactly across models which in many cases can be computationally difficult or intractable, resulting in the set of models which Thompson Sampling can be done for having limited representational capabilities. However methods such as Ensemble Sampling Lu and Van Roy (2017) propose a method for approximating ensemble sampling by tracking an ensemble of models, allowing an approximate version of Thompson Sampling to be applied more widely. Try adapting their method by modifying it to the contextual setting and running it on the dataset. Compare the performance of this algorithm with your Linear algorithm from the previous part. How does the performance change as the number of models in the ensemble increases? Given your results what can you say about how good a linear approximation is for this data?

- **Comparing to a Supervised Learning Baseline**: In the previous part we considered using the data to learn a policy using bandit learning algorithms. However we can compare to a "bandit" version of a supervised learning algorithm as well. To describe this algorithm, imagine starting with a randomly initialized supervised predictor $f_0(x)$ that can be of any function class of your choosing (for example linear regression). For every new datapoint x_i your bandit algorithm from the previous part sees, you can also first calculate the prediction of your supervised predictor on x_i, $f_{i-1}(x_i)$, to compute the performance metrics (*regret* and *the average fraction of incorrect dosing decisions*) for timestep i. After you can observe the true value of the dosage for x_i and refit f on all the data you have seen so far to obtain f_i. Compare the performance of this "bandit" supervised learning algorithm to your bandit algorithm from the previous part. Given that if you used a linear model for f and that f is an "oracle" in the sense that it is fit with the true values of dosage while your bandit algorithm was just given a

reward of 0 or -1, what does the performance of the "bandit" supervised learning algorithm say about the best possible performance of bandit algorithms? You can also try other models for the "bandit" supervised learning algorithm.

References

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

H. Bastani and M. Bayati. Online decision-making with high-dimensional covariates. 2015.

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.

X. Lu and B. Van Roy. Ensemble sampling. In *Advances in neural information processing systems*, pages 3258–3266, 2017.

B. Metevier, S. Giguere, S. Brockman, A. Kobren, Y. Brun, E. Brunskill, and P. S. Thomas. Offline contextual bandits with high probability fairness guarantees. In *Advances in Neural Information Processing Systems*, pages 14893–14904, 2019.

G. Neu and J. Olkhovskaya. Efficient and robust algorithms for adversarial linear contextual bandits. In *Conference on Learning Theory*, pages 3049–3068. PMLR, 2020.