Contextual Multi-Armed Bandits for Warfarin Dosing

William Bakst * **Nicholas Bien** * **Neel Ramachandran** *

*Equal contribution

Abstract

This paper implements and analyzes the usage of contextual multi-armed bandit (MAB) algorithms in prescribing Warfarin dosages to patients. We evaluate several contextual MAB algorithms and compare them to an offline learning baseline. Ultimately, we show that despite no prior knowledge or access to the dataset, online MAB algorithms performs as well as, and in some cases better than, the offline baseline.

1. Introduction

Warfarin is a commonly-used oral blood anticoagulant generally used to treat blood clots and prevent stroke. The specific dosage required by patient is difficult to establish because it ranges widely, from a few milligrams a week to over three hundred (based on the data used in this paper). As such, doctors typically prescribe an initial prescription to a patient and make adjustments to the dosage over time. Computational methods for determining proper Warfarin dosages also exist, yet most involve training a model on a large amount of pre-existing data (Consortium, 2009) before making accurate predictions. Both aforementioned approaches for Warfarin dosing depend on a "trial" period, where the agent (whether doctor or algorithm) gathers data before making a final recommendation. Here, we treat Warfarin dosing as an online learning decision-making problem, where we assume no prior dataset to train on, and give one recommendation without the ability to subsequently fine-tune. We model the Warfarin dosing as a contextual multi-armed bandit problem, where we choose a 'High', 'Medium', or 'Low' dosage for each patient we encounter and record our results over 5528 patients. We show that the multi-armed bandit approach outperforms pre-trained approaches such as linear regression and should be considered a viable approach to prescribing Warfarin dosing.

2. Dataset

We acquired a dataset of 5700 patients treated with Warfarin from the Pharmacogenetics and Pharmacogenomics Knowledge Base (PharmGKB) (Consortium, 2009). Patients in the dataset represent 9 countries across 4 continents. We filter our dataset to 5528 patients who have a non-null value of 'Therapeutic Dose of Warfarin', the variable we are interested in predicting. We bucket dosages as follows: less than 21 mg/week is considered 'Low'; between 21 and 49 mg/week is considered 'Medium'; more than 49 mg/week is considered 'High'. For a patient at time t, we build a 65-dimensional feature vector X_t. Among the variables we represent in the feature vector are basic patient characteristics (age, weight, height), genotypes (VKORC1, CYP2C9), general state of health (reason for treatment, comorbidities, diabetes), and medications (enzyme inducers, amiodarone). We treat each feature as a one-hot encoding including a bucket for missing values, such that X_t is a concatenation of one-hot encoded vectors representing each feature. We also perform imputations on VKORC1 values based on the procedure described in Appendix Section S4 of (Consortium, 2009).

3. Contextual Multi-Armed Bandit Framework

3.1. Modeling as a Bandit

As previously described, we model Warfarin dosing as an online-learning contextual multi-armed bandit (MAB) problem. The problem is formally described as follows: Given a set of actions $\mathcal{A}$ and a binary set of rewards $\mathcal{R} = \{0, 1\}$ at every timestep t we see some new contextual information X_t and make a decision $a_t \in \mathcal{A}$ such that $a_t = \arg\max_a P(R = 1|X_1, r_1 \dots X_t, X_{t-1}, r_{t-1}, a)$. We then 'pull' arm a_t which gives us reward r_t, and we use r_t and X_t to update our model. In the Warfarin problem setting, we have $\mathcal{A} = \{Low, Medium, High\}$, and feature vector X_t, which is a representation of patient attributes described in the previous section.

We implement three types of multi-armed bandits: the Linear Upper Confidence Bound (LinUCB), LASSO, and Thompson's Sampling, and finally apply the Multiplicative Weight Update method to LASSO and Thompson's. These methods are specified further in the following sections.

3.2. Evaluation

In the bandit setting, we evaluate our performance using (1) the fraction of incorrect predictions and (2) regret at each timestep t, where regret is defined as the sum of incorrect decisions. We evaluate these metrics over 20 instantiations of our data, shuffling the order of patients each time, to provide average scores with error bounds.

4. Related Work

4.1. LinUCB

(Chu et al., 2011) investigate the "multi-armed bandit problem with expert advice", in which given a sample, the bandit observes features for each action and chooses the best action given those features. They adapt the linear upper-confidence bound algorithm (LinUCB) to this contextual setting. The LinUCB algorithm computes a linear-regression based upper bound estimate of reward for each action and chooses the action with the highest upper bound. See Algorithm 1 or (Chu et al., 2011) for details. They prove a sublinear bound on cumulative regret for the contextual LinUCB algorithm with respect to time. We modify this algorithm for the setting of a single feature for all arms by learning a distinct regression for each arm.

4.2. LASSO

(Bastani & Bayat, 2015) improve on linear UCB by applying the LASSO estimator, which enforces regularization based on the L1-norm, to the bandit setting. They also construct forced-sample sets (timesteps where a preset action is taken regardless of the model's prediction) to encourage exploration. Actions are not taken if the predicted reward based on forced-sample set features falls below a threshold of the best action based on the forced-sample sets. They prove that the LASSO estimator still converges in the bandit setting where samples are not independent and identically distributed due to the updates of the learned policy after each sample.

4.3. Thompson

Thompson sampling is a well-established Bayesian solution the bandit problem. While Thompson sampling does not have as strong of bounds on performance compared to LinUCB, it often performs well in practice. It works by initializing a conjugate prior distribution (i.e. Gaussian if we want to sample from a Gaussian distribution or Beta if we want to sample from a Bernoulli distribution). We then sample from the posterior distribution, which we compute using the data (features and rewards) we have seen so far. (Agrawal & Goyal, 2013) provide theoretical guarantees for contextual Thompson sampling. Similar to our adaptation of (Chu et al., 2011) to the Warfarin problem, we adapt the Gaussian-based Thompson sampling algorithm for multi-feature samples to the case of a single feature by learning different parameters for each arm.

5. Approach

Here we outline the algorithms that we implemented. We discuss the changes we made to better suit our problem setting as well as outline fully these updated algorithms.

5.1. Baselines

5.1.1. Single Action

This baseline simply chooses a medium dosage for all patients, mimicking the approach commonly-used by doctors. This algorithm performs much better than expected because of the class imbalance in the dataset: 61.2% of patients receive a dosage in the medium bucket.

5.1.2. Linear Regression

We use linear regression model as a second baseline model. This model learns weights for a subset of features trained on 80% of the patients available; here, we simply use the weights provided by the "Warfarin Clinical Dosing Algorithm" in Appendix Section S1f of (Consortium, 2009). We also filter the dataset to 4386 patients whose values are non-null for the features described in the paper. Note that we then report the accuracy of the model on 100% of the patients available, which overestimates the actual accuracy of the model but provides a target 'oracle' that represents how well one can expect to do in an online setting.

5.2. Linear UCB

Although the Linear UCB algorithm described in (Chu et al., 2011) was designed for the same Warfarin dataset, the original algorithm uses features for each arm, whereas we use features for each patient. We update the algorithm such that we calculate a different matrix A_i and vector b_i for each arm and only update the A_i and b_i of the arm i selected for each patient.

5.3. LASSO Bandit

We simplify LASSO with modified forced sampling that encourages early exploration without keeping a second set of weights to narrow down action class. We found that since the action class is small (size 3), excluding actions based on a few samples was noisy and we were better off just choosing the arm with highest expected reward given all the data so far. We also found that using rewards 0 for correct prediction and -1 outperformed using reward 1 and 0, respectively, which is what worked best for the other

Algorithm 1 LinUCB

Input : α, K, d
Initialize A_i as I_d and b_i as $\mathbf{0}_d$ for each $i \in K$
for $p \in$ PATIENTS **do**
  Observe X_p feature vector for patient p
  $\theta_{i,p} \leftarrow A_i^{-1} b_i$ for each $i \in K$
  for $a \in K$ **do**
    $prob_{a,p} \leftarrow \theta_{a,p}^T X_p + \alpha\sqrt{X_p^T A_a^{-1} X_p}$
  end for
  Play arm $a_p \leftarrow \arg\max_a prob_{a,p}$
  Observe reward r_p
  Update $A_{a_p} \leftarrow A_{a_p} + X_p X_p^T$
  Update $b_{a_p} \leftarrow b_{a_p} + X_p r_p$
end for

models. We set $q = 3$, $h = 1$, $\lambda_1 = \lambda_2 = 0.05$.

Algorithm 2 LASSO

Input : $q, h, \lambda_1, \lambda_{2,0}$
Initialize $\mathcal{T}_{i,0}$ and $\mathcal{S}_{i,0}$ as empty sets, $\hat{\beta}(\mathcal{T}_{i,0}, \lambda_1)$ and $\hat{\beta}(\mathcal{S}_{i,0}, \lambda_{2,0})$ as $\mathbf{0} \in \mathbb{R}^d$ for all $i \in K$
Construct forced sample sets (Bastani & Bayat, 2015)
for $p \in$ PATIENTS **do**
  Observe X_p feature vector for patient p
  if $p \in T_i$ for any i **then**
    $a_p \leftarrow i$
  else
    $\hat{\mathcal{K}} = \Big\{ i \in K \;\Big|\; X_P^T \hat{\beta}(\mathcal{T}_{i,p-1}, \lambda_1) \geq \max_{j \in K} X_p^T \hat{\beta}(\mathcal{T}_{j,p-1}, \lambda_1) - h/2 \Big\}$
    $a_p \leftarrow \arg\max_{i \in \hat{\mathcal{K}}} X_p^T \hat{\beta}(\mathcal{S}_{i,p-1}, \lambda_{2,p-1})$
  end if
  $\mathcal{S}_{a_p,p} \leftarrow \mathcal{S}_{a_p,p-1} \cup \{p\}$, $\lambda_{2,p} \leftarrow \lambda_{2,0}\sqrt{\dfrac{\log p + \log d}{p}}$
  Play arm a_p, observe reward r_p
end for

5.4. Thompson Sampling

The Thompson sampling method described in (Agrawal & Goyal, 2013) assumes a different feature for each action for each sample in the dataset. We modify this by storing B, f and $\hat{\mu}$ parameters for each arm. We use $v = 0.25$ as the standard deviation multiplier for the multivariate normal sampling.

5.5. Adaptation of Multiplicative Weight Update

The original multiplicative weight update algorithm was designed for probabilistically selecting an arm based on a number of "expert" decisions. Here we update the algorithm such that each "expert" is instead an initialization of a Thompson Sampler, and we select the arm that receives the highest weighted vote among these samplers. We update the weight of each samplers vote after each patient depending on whether or not that particular sampler selected the correct arm. We designed this algorithm to help stabilize the results of our Thompson Sampler. Since Thompson Sampling samples actions from a determined probability distribution, there is always a chance that it samples the incorrect arm. However, by using multiple Thompson samplers and weighting their votes, we are much less likely to choose the incorrect arm because the chances that all arms choose the wrong arm with low probability is very low. Thus MWU will choose the arm with highest probability more often. We can think of this as a tradeoff between exploitation and exploration as we see more patients since early on we have equally weighted initial samplers and then both weight samplers that are correct more often as well as increase the chances that we choose the arm with the highest overall probability of being correct.

Algorithm 3 Thompson Sampling

Input : K, d, v
Initialize B as I_d, $\hat{\mu}$ as $\mathbf{0}_d$ and $f = \mathbf{0}_d$ for each $i \in K$
for $p \in$ PATIENTS **do**
  Observe X_p feature vector for patient p
  for $a \in K$ **do**
    Sample μ_t from distribution $\mathcal{N}(\hat{\mu}_a, v^2 B_a^{-1})$
    $prob_a \leftarrow X_p^T \mu_t$
  end for
  Play arm $a_t \leftarrow \arg\max_a prob_a$
  Observe reward r_t
  Update $B_{a_t} \leftarrow B_{a_t} + X_p X_p^T$
  Update $f \leftarrow f + X_p r_t$
  Update $\hat{\mu} \leftarrow B^{-1} f$
end for

Algorithm 4 MWU

Input : K, N, η
Initialize N Thompson Samplers $T \leftarrow \{T_1, ..., T_N\}$
Initialize weights $w^{(0)} \leftarrow \{w_1^{(0)}, ..., w_N^{(0)}\}$
for $p \in$ PATIENTS **do**
  $v_p \leftarrow w^{(p-1)} / \sum_{i \in N} w_i^{(p-1)}$
  Observe choices a_i for each $T_i \in T$
  $A_k \leftarrow \sum_{i:a_i=k} v_{p,i}$ for each $k \in K$
  Play arm $a = \arg\max_k A_k$
  Observe reward r_p
  Update parameters for each $T_i \in T$
  $w_i^p \leftarrow w_i^{p-1} * \eta^{-r_p}$
end for

Table 1. Accuracy for each approach

APPROACH	ACCURACY
SINGLE ACTION	0.6118 ± 0.0000
LINEAR REGRESSION	0.6536 ± 0.0000
LINUCB	0.6470 ± 0.0019
LASSO	0.6507 ± 0.0029
THOMPSON	0.6577 ± 0.0019
MWU	0.6596 ± 0.0033

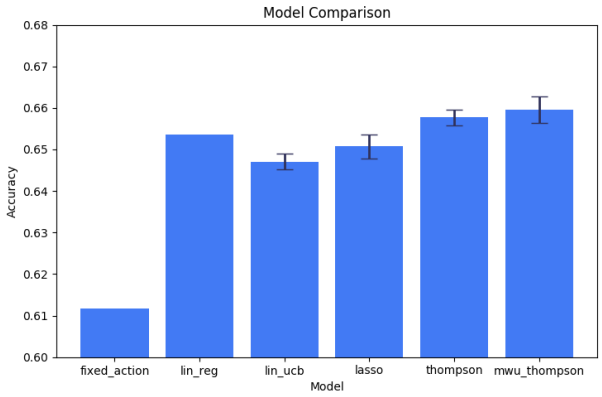

Figure 1. Overall accuracy with 95% confidence intervals obtained by evaluating on 20 random permutations of the dataset. Fixed action refers to prescribing the medium dosage to every patient. Linear regression uses coefficients from Section S1f of the appendix in (Consortium, 2009), which were trained on 80% of the dataset and evaluated on 20%. We evaluate linear regression on the whole dataset for consistency with our other results. Thompson sampling and Thompson sampling with multiplicative weight updates demonstrate statistically significant improvements over the linear regression baseline.

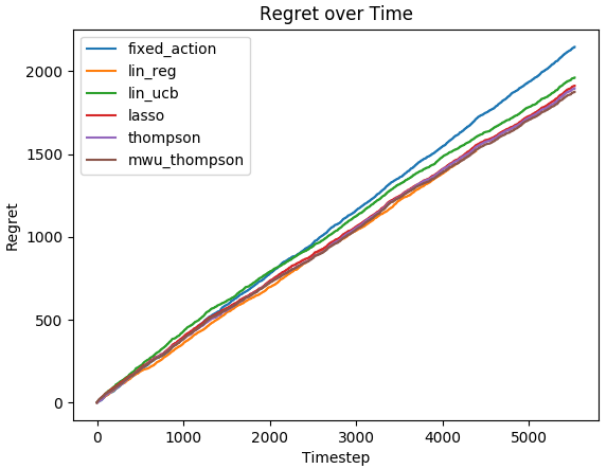

Figure 2. Regret at each iteration on a random permutation of the dataset.

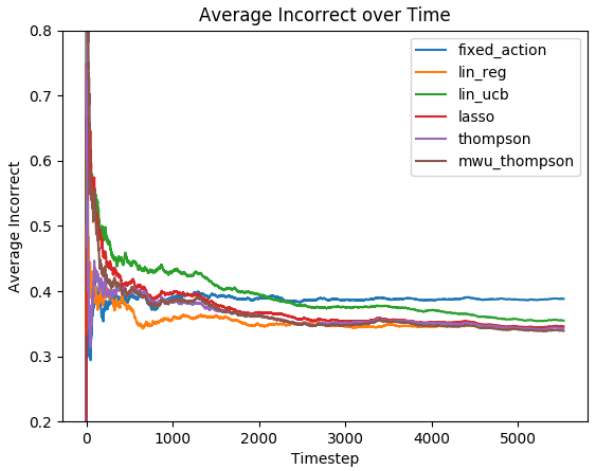

Figure 3. Fraction of incorrect predictions at each iteration on a random permutation of the dataset.

6. Experimental Results

First, we see that Linear UCB and LASSO both perform worse than linear regression but better than the single action baseline by the 2000th timestep(Table 1; Figure 1). Since Linear Regression uses the optimal learned weights and both LinUCB and Lasso are essentially doing linear regression in the online setting, so we do not expect them to do better than linear regression, but we do expect it to learn something and do better than just single action.

We can also see that Thompson Sampling performs better than linear regression. This is likely because we have expanded our feature vector and because using Bayesian priors allow us to model uncertainty for better decision making in the early stages of training. We can also see that using MWU stabilizes the results at a higher value since using multiple samplers and weighting the votes helps improve the probability that we select the correct action even if one or more samplers individually make mistakes. We also never perform worse than the worst Thompson sampler, and our confidence interval reaches a higher accuracy. We were surprised to find that multiplicative weight updates increased variance. We hypothesize that this might be due to adding more randomness by using 10 Thompson samplers. It is also notable that adding MWU on top of Thompson makes our estimates more "conservative": high doses are prescribed less often while medium doses are prescribed more often. This may be due to more exploratory experts being down-weighted in voting because they get more samples incorrect in the early stages of training.

All Bandit algorithms begin to outperform the single action baseline after about 2000 iterations (Figure 3). Lasso and Thompson sampling reach linear regression's fraction incorrect faster than linear UCB; however, LinUCB appears to

Table 2. Confusion Matrix (averaged over 5 random permutations of the dataset). Bolded in blue are the percentage of patients who were dosed correctly; bolded in red are the patients who were dosed incorrectly by two buckets (stratified by true dosage).

		LINEAR REGRESSION			LINUCB			LASSO			THOMPSON'S			MWU		
		Low	*Medium*	*High*	*Low*	*Medium*	*High*	*Low*	*Medium*	*High*	*Low*	*Medium*	*High*	*Low*	*Medium*	*High*
TRUE DOSAGE	*Low*	**36%**	64%	**< 1%**	**50%**	48%	**2%**	**28%**	72%	**< 1%**	**43%**	57%	**< 1%**	**39%**	61%	**< 1%**
	Medium	10%	**89%**	1%	14%	**79%**	7%	7%	**92%**	< 1%	10%	**88%**	2%	9%	**91%**	< 1%
	High	**< 1%**	90%	**10%**	**2%**	75%	**22%**	**< 1%**	99%	**< 1%**	**3%**	88%	**9%**	**2%**	98%	**< 1%**

be improving at a faster rate on later examples. UCB has stronger theoretical convergence guarantees than Thompson sampling. Thompson outperforms UCB on this small dataset, but UCB appears to has a more sublinear regret curve (Figure 2). It is unclear how these models would compare given more samples.

Table 2 shows the confusion matrices for each our models compared to the linear regression baseline. This shows that Lasso and MWU are more "conservative", rarely predicting a high dosage, while LinUCB is the most well-balanced, predicting the highest percentage of high dosages correctly, followed by Thompson sampling, which has a distribution of predictions similar to linear regression. We can see also see that all our models are well-calibrated in that they very rarely predict a dosage that is two buckets away from the true dosage, (ie, a high dosage when the true dosage is low, or a low dosage when the true dosage is high). Based on the confusion matrices, LinUCB and Thompson sampling appear to offer the best balance between accuracy and diversity of predictions, though due to the target class imbalance, MWU Thompson sampling achieves the highest overall accuracy.

7. Conclusion

We analyzed contextual multi-armed bandit (MAB) algorithms in prescribing Warfarin dosages to patients. We evaluated four models–linear UCB, LASSO, Thompson sampling, and Thompson sampling with multiplicative weight updates–in comparison to a fixed-action baseline and a linear regression oracle. On a dataset of Warfarin doses with patient health data as features, we found that despite no prior knowledge or access to the dataset, online MAB algorithms performs as well, and in the case of Thompson sampling and MWU Thompson sampling, better than the offline baselines.

We note that Thompson sampling with multiplicative weight updates achieves the highest accuracy (and lowest regret) among our contextual MAB baselines. Nevertheless, it is worth noting that our evaluation methods (accuracy and regret) are defined in terms of the number of correct predictions, but may not be entirely indicative of positive impact on Warfarin dosing. It is also important to understand the implications of predicting a 'medium' dosage when the true dosage is 'low', for example. Understanding these tradeoffs requires more domain knowledge, which we leave to the science experts.

This work shows that bandit algorithms could potentially be utilized in healthcare without any initial training data and provide diagnoses that are comparable, if not better, than current practices, making them valuable in contexts where data is scarce.

8. Code

We have created a public GitHub repository containing implementations of the previously described algorithms. You can find the code at:

https://github.com/wbakst/WarfarinBandits

References

Agrawal, S. and Goyal, N. Thompson sampling for contextual bandits with linear payoffs. In *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ICML'13, pp. III–1220–III–1228. JMLR.org, 2013. URL http://dl.acm.org/citation.cfm?id=3042817.3043073.

Bastani, H. and Bayat, M. Online decision-making with high-dimensional covariates. SSRN, 2015. URL http://dx.doi.org/10.2139/ssrn.2661896.

Chu, W., Li, L., Reyzin, L., and Schapire, R. Contextual bandits with linear payoff functions. In Gordon, G., Dunson, D., and Dudk, M. (eds.), *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pp. 208–214, Fort Lauderdale, FL, USA, 11–13 Apr 2011. PMLR. URL http://proceedings.mlr.press/v15/chu11a.html.

Consortium, I. W. P. Estimation of the warfarin dose with clinical and pharmacogenetic data. *New England Journal*

of Medicine, 360(8):753–764, 2009. URL https://doi.org/10.1056/NEJMoa0809329.