Estimation of Warfarin Dosage with Reinforcement Learning

Cedrick Argueta[1], Nguyet Minh Phu[1], Emanuel Pinilla[1]

[1]*Department of Computer Science, Stanford University, Stanford, CA 94305*

Motivation

Warfarin is a popular oral blood anticoagulant agent. It is important to prescribe the correct warfarin dosage to avoid the severe adverse effects of taking an incorrect dose. Determining the correct dosage of warfarin is challenging, however, because the dosage varies significantly among patients. This project aims to use various reinforcement learning approaches to build the best model for ***Warfarin dosage prediction***.

Data

We use the *PharmGKB dataset* consisting of information for 5700 patients treated with Warfarin. The dataset consists of information about each patient, including gender, race, ethnicity, age, height, and so on, as well as the correct Warfarin dosage. We drop 173 patients without information on the correct Warfarin dosage.

Missing Data Values of age, height and weight are missing for some patients. We fill in the missing ages with the mode of all other ages, and the missing heights and weights with the average of all other heights and weights respectively. For all other features, we treat "missing" as an additional possible value.

Feature Engineering To represent each patient, we use a combinations of 26 features that include:

- Age in decades
- Height in cm
- Weight in kg
- Race (Indicators of is_Asian, is_Black, is_Missing)
- Enzyme Inducer Status (whether the patient is taking Carbamazepine, Phenytoin, Rifampin or Rifampicin)
- Whether the patient is taking Amiodarone
- Gender
- VKORC1 genotype
- VKORC1 QC genotype
- CYP2C9 consensus

References

- Li, Lihong et al. "A Contextual-Bandit Approach to Personalized News Article Recommendation."
- I. W. P. Consortium. "Estimation of the warfarin dose with clinical and pharmacogenetic data."

Methods

Baselines

- Choose medium dosage for all patients
- Choose based on linear combination of some features

Warfarin clinical dosing algorithm		
	4.0376	
-	0.2546 x	Age in decades
+	0.0118 x	Height in cm
+	0.0134 x	Weight in kg
-	0.6752 x	Asian race
+	0.4060 x	Black or African American
+	0.0443 x	Missing or Mixed race
+	1.2799 x	Enzyme inducer status
-	0.5695 x	Amiodarone status
=	Square root of weekly warfarin dose**	

LinUCB

An improvement is to choose an 'arm' (low, medium, high dose) based on its upper confidence bound for reward.

$$a_t = argmax_{a \in A_t} \Big(\underbrace{\mathbf{x}_{t,a}^\top \hat{\theta}_a}_{\text{predicted payoff}} + \alpha \underbrace{\sqrt{\mathbf{x}_{t,a}^\top \mathbf{A}_a^{-1} \mathbf{x}_{t,a}}}_{\text{standard deviation of payoff}} \Big)$$

Reward Shaping

The default rewards in {0, 1} don't encode for when one failure is worse than another, .e.g. giving a low dose to a patient who needs a high dose vs. giving a high dose to a patient who needs a low dose.

We use this reward structure to fix this, with R = 1.5.

Given > Required v	Low	Medium	High
Low	0	-R / 2	-2 * R
Medium	-R / 2	0	-R / 2
High	-R	-R / 2	0

Online Supervised Learning w/ Linear Regression

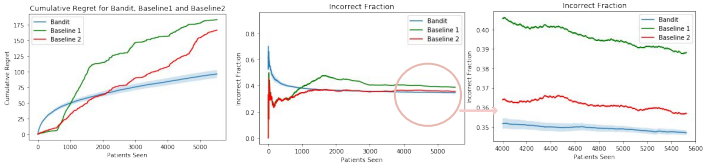
We also train a supervised learning (SL) agent in a bandit fashion, fitting a linear regression (LR) model on all previously seen data whenever a prediction must be made. We perform regression directly on the dose required rather than the (low, medium, high) buckets, so this model is stronger than the ones mentioned above.

Online Supervised Learning w/ Neural Networks

With this agent, we must train a NN to convergence after seeing each new patient. This method, while allowing for more powerful function approximators than LR, is much slower because of an inner loop for training. After implementing this, it was too slow to compare against other methods, but can potentially be stronger than LR.

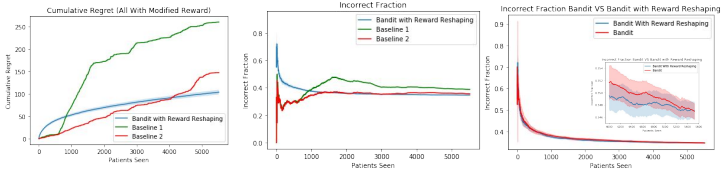
Results

Baselines & LinUCB



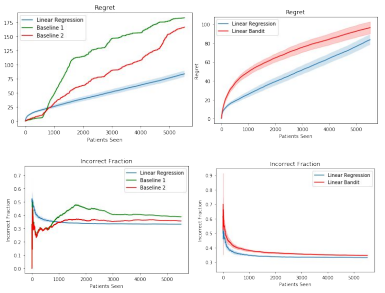
We can see that our LinUCB implementation has sublinear regret, and clearly beats both baselines in overall accuracy in predictions. The third graph is a zoomed in version of the second graph.

Reward Reshaping

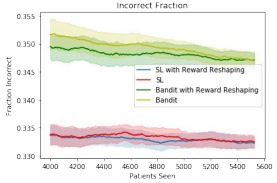


Reward shaping improves accuracy with the LinUCB algorithm, but changes our regret graphs because of the new reward values.

Online Supervised Learning w/ Linear Regression



Online Supervised Learning w/ Linear Regression and Reward Reshaping



Combining Reward Reshaping and Linear Regression significantly outperforms LinUCB.

The online linear regression models are upper bounds on accuracy of LinUCB, because they perform regression directly on required dose rather than the correct bucket. Using the dosage directly provides more information than using buckets. Online linear regression is significantly better than both of our baselines.

Conclusion

- LinUCB is superior to both baselines.
- Linear regression on the actual dosage gives us a stronger predictor than LinUCB with the buckets, so it's an upper bound on the performance of LinUCB.
- Reward shaping helps with improving accuracy with all models, though only marginally.
- Future work includes comparison with neural network in online supervised learning.