Contents

1

Questions

WHAT IS THE PRICE of personalization? We work with HeartSteps v1 data in order to choose an online learning algorithm for future HeartSteps trials. This project aims to answer the question on whether it is possible to tailor just-in-time intervention messages to individual users in a way that does not jeopardize the results of other users.

We choose to answer the following questions (see Table 3.1 for notation):

1. Consider multiple simulated users, all with the same state distribution and the same reward distribution. How does the achieved reward vary by user? How much variance is there from user to user?

   Can compare this mean/variance to the mean/variance in reward across users if treatment is randomized with probability 0.6.

2. If there is no treatment effect (i.e. the generative model in the simulation for the reward does not depend on action $\mathcal{A}$, or that the true value of interaction terms of Θ are 0), then what doe the selection probabilities do? Does the choice of the prior mean, μ_Θ affect this? For example, for the interaction terms, if μ_Θ is not close to 0 but in reality true $\Theta = 0$.

3. How fast does the bandit algorithm result in action selection probabilities equal to 0.2 or 0.8? How does this vary with SNR? How does this vary with the diagonal variance terms in Σ_Θ, the prior on Θ?

4. How robust is the bandit algorithm to misspecification of linear model for the true generative model given context? How does this vary with signal to noise ratio (Equation 3.4)?

5. How sensitive is the bandit algorithm to really good initialization of the Θ coefficients in the linear model? that is, when the prior on Θ has a mean μ_Θ that is close or equal to the true values of Θ in the true reward generative model. How sensitive is the bandit algorithm to really poor initialization of the coefficients in the linear model?

6. If the true underlying model for the conditional mean of the reward given context varies with time t in a way that leads to differing optimal actions from some commonly occurring contexts, then does the bandit algorithm adjust to this?

This is some random quote to start off the chapter.

Firstname lastname

2

Data

LOREM IPSUM DOLOR SIT AMET, consectetuer adipiscing elit.

Description of Data

Describe rewards, contextual features

3

Methods

Throughout this project, we work with the HeartSteps v1 Data, hereupon abbreviated HSv1. Features have been created from the measurements through domain science, through which we assume that the Bandit algorithms use linear models for the purposes of the project.

The contextual bandit algorithms used are stochastic, meaning for every user n, day t, and decision point t, the algorithm generates a probability $\pi_{n,t,d}$ of action. Thus, whethere an action is taken is generated by

$$A_{n,t,d} \sim \text{Bern}(\pi_{n,t,d}).$$

The main metric of performance on a user we use is Mean User Expected Reward, or *MUER*. This is computed in equation 3.1 using the reward $R^{(a)}_{n,t,d}$ of either action $a = 0$ or $a = 1$ under our generative model:

$$MUER(\mathcal{S}, \Theta, \varepsilon, n) = \frac{1}{TD} \sum_{\substack{t=1,\ldots,T \\ d=1,\ldots,D}} \pi_{n,t,d} R^{(1)}_{n,t,d} + (1 - \pi_{n,t,d}) R^{(0)}_{n,t,d}, \tag{3.1}$$

$$R^{(a)}_{n,t,d} = \begin{bmatrix} f_1(S_{n,t,d}) \\ a \cdot f_2(S_{n,t,d}) \end{bmatrix}^T \Theta + \epsilon_{n,t,d}, \tag{3.2}$$

Note that we add in $\epsilon_{n,t,d}$ to better account for misspecification in our 'true' generative model.

For each simulation of N users, we can now compute the mean and standard deviation of *MUER* as our main performance metrics.

3.1 Methodology Overview

For simulations, our overall methodology is the following. We first designate a variant of the Bandit algorithm, as well as a 'true' generative model, using f_1, f_2, to form simulated rewards from the context and given action. Next, we split HSv1 user data into K-fold cross validation sets. For each split, we perform the below process:

1. Use the training and testing data to generate training simulated users and testing simulated users, with $N = 500$ users in simulation.
2. Use training simulated users to tune parameters of the given Bandit algorithm based on the mean of *MUER* across all users and decision points, contingent on quality metrics.
3. Run simulated users from test data using tuned parameters, observing quality metrics and impact of each tuning parameter on mean and std *MUER*.

Each part is described in more detail below.

3.2 Reward Generative Model

We set different 'true' generative models, where we assume:

$$\mathcal{R} = \begin{bmatrix} f_1(\mathcal{S}) \\ \mathcal{A} \odot f_2(\mathcal{S}) \end{bmatrix}^T \Theta + \varepsilon \tag{3.3}$$

Table 3.1: Notations

Term	Name	Description
$\mathcal{S}, S_{n,t,d}$	Context	Set of 7 features
p_1	Baseline features dimension	Full set consists of 7 features with 1 bias term, Small set consists of 2 with 1 bias term
p_2	Interaction features dimension	Full set consists of 3 features with 1 bias term, Small set consists of 2 with 1 bias term
$\mathcal{A}, a_{n,t,d}$	Actions	Binary – 1: active message sent, 0: no active message sent
$\mathcal{R}, R_{n,t,d}$	Reward	Log-transformed step count in 30 minutes following decision point
$f_1 : \mathcal{S} \to \mathbb{R}^{p_1}$	Baseline feature mapping	Maps context to baseline features
$f_2 : \mathcal{S} \to \mathbb{R}^{p_2}$	Interaction feature mapping	Maps context to interaction features, which are multiplied by $\mathcal{A}$
Θ	'True' generative model coefficients	From regression on HSv1 data: $\mathcal{R} \sim [f_1(\mathcal{S}), f_2(\mathcal{A} \cdot \mathcal{S})]^T\Theta$
$\mathcal{E}, \varepsilon_{n,t,d}$	Linear model residuals	Residuals from HSv1 data after regression
N	Number of users	$N = 37$ in HSv1; $N = 500$ in simulations
T	Number of days in study	$T = 42$
D	Number of decision points per day	Set to $D = 5$
K	Number of cross-validations per test	Set to $K = 3$

and that f_1, f_2 are feature functions from our set of features $\mathcal{S}$ to baseline and interaction terms, $\odot$ denotes the term-wise product, and each $\varepsilon \sim \mathcal{N}(0, \sigma^2)$ for σ^2, which is a tuning parameter with estimator $\hat{\sigma^2} = \mathrm{Var}(\varepsilon)$.

MOVE THIS PART Recall that the Signal-to-Noise-Ratio (SNR) is computed as

$$\frac{\mathrm{Var}(\mathcal{R})}{\sigma^2} = \frac{\mathrm{Var}\left(\begin{bmatrix} f_1(\mathcal{S}) \\ A \odot f_2(\mathcal{S}) \end{bmatrix}^T \Theta\right)}{\sigma^2}. \tag{3.4}$$

We describe these models more in Section 4.1.

3.3 Cross-Validation

We use $K = 3$ fold cross-validation to separate training and testing batches. We do not train the bandit algorithm's parameters on the test batches, as to simulate application of HSv1 data for HSv2.

To do this, we randomly order the $N = 37$ users, then group the randomly ordered users into K groups; the K-th fold cross-validation is performed holding the K-th group out as the test sample, and the remaining groups in as the training sample.

For each user, we have a series across all days T and decision points t per day of Reward, Action, and Context. We will refer to them jointly as $(R, A, S)_{(N,T,t)}$, which are implicitly indexed in order by the user, day, and decision point. Thus, there are $N \times T \times t$ data points.

3.4 Residual Formation

Within each test batch and train batch, we conduct ordinary least squares linear regression to residualize additional effects, missing from our model due to possible misspecification, for each user from the baseline and interaction effects. Specifically, we use the model in Equation 3.3.

Recall that f_1, f_2 are the baseline and interaction functions, that are parameters of the generative model.

We thus obtain a 'true' Θ for the simulation as well as a series of ε for each data point.

3.5 Simulated User Generation

For both the training and test original users' series of $(\mathcal{R}, \mathcal{A}, \mathcal{S})$, we can create train and test batches of $N = 500$ users each by randomly sampling with replacement from the train and test pools respectively. Picking a high N for simulations gives statistical significance in our stochastic algorithm.

We note finally that in each of the $N = 37$ original HSv1 users, we imputed the mean value for missing features, and set availability to True when the reward and at least one of the features is measured.

3.6 Training Bandit Algorithm Tuning Parameters

For each variant of the Bandit Algorithm, we will tune parameters differently. These will be addressed in Section 4.2.

Ultimately, the models will be tuned according to minimizing the mean *MUER*, subject to reasonable quality metrics and *MUER* standard deviation.

3.7 Simulated User Testing and Quality Metrics

Once the tuning parameters have been optimized for the batch, we run the test users on with these parameters, and analyze the quality metrics described below:

1. Examine time series of $\pi_t(1|S_t)$, the probability of taking action 1 for all t, looking through several users.
2. Examine $\pi_t(1|S_t)$ vs $opt_t(S_t)$ for all t, looking through several users.

   We define $opt_t(S_t)$ as the optimal probability in context S_t:

$$opt_t(S_t) = 0.8\,1\{\text{optimal action is 1 in } S_t\} + 0.2\,1\{\text{optimal action is 0 in } S_t\}. \quad (3.5)$$

3. Examine $|\pi_t(1|S_t) - opt_t(S_t)|$ for all users, averaging over all N users for each time point t.
4. Examine $|\pi_t(1|S_t) - opt_t(S_t)|$ for all users, plotting histogram of each user's mean
5. Examine cumulative regret over t, plotting average over all users as well as for several individual users.

   Cumulative regret is the expected reward of the bandit minus reward of the optimal policy.

6. Examine the number of actions taken at each time t, plotting histogram across all t for each of several simulated users, as well as the average taken across all N users plotted at each time t.

Algorithm 1: Simulated User Generation Pseudocode

Data: $(\mathcal{S}, \varepsilon)^{batch}_{N,T,D}$

Result: $(\mathcal{S}, \varepsilon)^{sim}_{N,T,D}$

for $1 \leq n \leq N_{new}$ do
    while $(\mathcal{S}, \varepsilon)^{batch}_{n,:,:}$ *does not have* N_{sim} *data points* do
        Sample a single user $n' \in [1, N_{batch}]$ without replacement;
        Append $(S, \varepsilon)^{batch}_{n',:,:}$ to $(S, \varepsilon)^{sim}_{n,:,:}$;
    end
end

4

Models

Several different variants of the Multi-armed Contextual Bandit algorithm were investigated to test feasibility in the HeartSteps mobile application.

4.1 Reward Generative Models

4.1.1 Basic Linear Generative Model

The most basic generative model for rewards is based on the equation:

where we set $p_1 = 8, p_2 = 3, f_1 : \mathcal{S} \to \mathbb{B}^{p_1}$ to be the identity of the 7 features plus a bias feature, and $f_2 : \mathcal{S} \to \mathbb{R}^{p_2}$ to be the identity on the first 3 features.

4.1.2 Additional Models

Can test time-varying reward functions, or perhaps some non-identity/non-linear functions?

4.2 Bandit Algorithm Variants

Can also use this section to describe motivation for each part of the Bandit algorithm.

We currently are using Peng's Algorithm 2 (Kristjan's Bandit Algorithm for HS2 (Action-Center Version), which contains a Gaussian Process, a Feedback Controller based on recent dosage, probability clipping, with action centering on the Gaussian Process update.

We plan on including/excluding each of the 4 modifications above (for $2^4 = 16$ slightly different variants), checking whether we need to include them or not.

There are the following parameters to optimize:

- Gaussian Prior parameters $(\gamma, \mu_\Theta, \Sigma_\Theta)$ (Peng uses μ_β, Σ_β instead)

- Feedback controller parameters (λ_c, N_c, T_c), where N_c is the desired dosage (number of 1 actions) over the past T_c decision times, and λ_c is the coefficient on how powerful the controller is

- σ^2, an estimate of the reward noise variance

- Probability clipping $(\pi_{\min}, \pi_{\max})$. We set these to $(0.2, 0.8)$ from domain science, that we require some amount of randomization.

- Baseline Features $f_1 : \mathcal{S} \to \mathbb{R}^{p_1}, f_2 : \mathcal{S} \to \mathbb{R}^{p_2}$. We set these to identity functions, where f_1 gives a bias term plus the original 7 context features, and f_2 gives the first 3 features for interaction terms.

We aim to tune the parameters in the above order.

1. Tune γ through parameter sweep on γ.

2. Tune Σ_Θ setting it to $v\mathbb{I}$, where $v \in \mathbb{R}$ is a scalar and $\mathbb{I}$ is the identity matrix; parameter sweep on v.

3. Set $\Sigma_\Theta = \Theta$ from the training data regression.

4. Tune λ_c through parameter sweep on λ_c.

5. Tune T_c through parameter sweep on T_c. Will stick to some discrete values such as $(5, 10, 50, 70)$ to not overfit.

6. Tune N_c setting it to mT_c, where T_c was the optimal value from the previous tuning step, and m is a scalar, likely in the range $(0.25, 0.75)$.

7. Set σ^2 to $\hat{\sigma^2}$, the empirical residual (noise) variance.

8. Set $\pi_{\min} = 0.2$, $\pi_{\max} = 0.8$.

9. Set f_1, f_2 to identity mappings.

Quick Note: on test data (i.e. random $\mathcal{S}$ and $\Theta = 100 \cdot \text{range}(11)$, and small Gaussian noise), the implemented Bandit Algorithm works very well to learn the true Θ when no action-centering occurs; otherwise, 3 coefficients in Θ are thrown off by subtracting π_t from A_t on line 26 in the algorithm, but the remainder are perfectly fine.

5

Results

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Morbi commodo, ipsum sed pharetra gravida, orci magna rhoncus neque, id pulvinar odio lorem non turpis. Nullam sit amet enim. Suspendisse id velit vitae ligula volutpat condimentum. Aliquam erat volutpat. Sed quis velit. Nulla facilisi. Nulla libero. Vivamus pharetra posuere sapien. Nam consectetuer. Sed aliquam, nunc eget euismod ullamcorper, lectus nunc ullamcorper orci,

fermentum bibendum enim nibh eget ipsum. Donec porttitor ligula eu dolor. Maecenas vitae nulla consequat libero cursus venenatis. Nam magna enim, accumsan eu, blandit sed, blandit a, eros.

6 Discussion

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Morbi commodo, ipsum sed pharetra gravida, orci magna rhoncus neque, id pulvinar odio lorem non turpis. Nullam sit amet enim. Suspendisse id velit vitae ligula volutpat condimentum. Aliquam erat volutpat. Sed quis velit. Nulla facilisi. Nulla libero. Vivamus pharetra posuere sapien. Nam consectetuer. Sed aliquam, nunc eget euismod ullamcorper, lectus nunc ullamcorper orci,

fermentum bibendum enim nibh eget ipsum. Donec porttitor ligula eu dolor. Maecenas vitae nulla consequat libero cursus venenatis. Nam magna enim, accumsan eu, blandit sed, blandit a, eros.

7

Conclusion

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Morbi commodo, ipsum sed pharetra gravida, orci magna rhoncus neque, id pulvinar odio lorem non turpis. Nullam sit amet enim. Suspendisse id velit vitae ligula volutpat condimentum. Aliquam erat volutpat. Sed quis velit. Nulla facilisi. Nulla libero. Vivamus pharetra posuere sapien. Nam consectetuer. Sed aliquam, nunc eget euismod ullamcorper, lectus nunc ullamcorper orci,

fermentum bibendum enim nibh eget ipsum. Donec porttitor ligula eu dolor. Maecenas vitae nulla consequat libero cursus venenatis. Nam magna enim, accumsan eu, blandit sed, blandit a, eros.