Reward Design for an Online Reinforcement Learning Algorithm Supporting Oral Self-Care

Anna L. Trella[1], Kelly W. Zhang[1], Inbal Nahum-Shani[2], Vivek Shetty[3], Finale Doshi-Velez[1], Susan A. Murphy[1]

[1] Department of Computer Science, Harvard University
[2] Institute for Social Research, University of Michigan
[3] Schools of Dentistry & Engineering, University of California, Los Angeles
annatrella@g.harvard.edu, kellywzhang@seas.harvard.edu, inbal@umich.edu, vshetty@ucla.edu, finale@seas.harvard.edu, samurphy@fas.harvard.edu

Abstract

While dental disease is largely preventable, professional advice on optimal oral hygiene practices is often forgotten or abandoned by patients. Therefore patients may benefit from timely and personalized encouragement to engage in oral self-care behaviors. In this paper, we develop an online reinforcement learning (RL) algorithm for use in optimizing the delivery of mobile-based prompts to encourage oral hygiene behaviors. One of the main challenges in developing such an algorithm is ensuring that the algorithm considers the impact of current actions on the effectiveness of future actions (i.e., delayed effects), especially when the algorithm has been designed to run stably and autonomously in a constrained, real-world setting characterized by highly noisy, sparse data. We address this challenge by designing a quality reward that maximizes the desired health outcome (i.e., high-quality brushing) while minimizing user burden. We also highlight a procedure for optimizing the hyperparameters of the reward by building a simulation environment test bed and evaluating candidates using the test bed. The RL algorithm discussed in this paper will be deployed in Oralytics. To the best of our knowledge, Oralytics is the first mobile health study utilizing an RL algorithm designed to prevent dental disease by optimizing the delivery of motivational messages supporting oral self-care behaviors.

1 Introduction

Dental disease is as common as it is preventable in the United States (Benjamin 2010). Apart from causing pain and imposing financial costs for treatment, oral health problems affect people's ability to eat, swallow, speak and socialize; they also increase the risk of complications. Chronic oral infections have been associated with diabetes, heart and lung disease, stroke, and low birth weight. Although dental disease is preventable through systematic, twice-a-day tooth brushing (Löe 2000), this behavior is not widely practiced because patients forget or abandon clinician instructions (Martin et al. 2005).

Ultimately, oral health depends on the individual's ability and willingness to carry out consistent brushing behaviors. As innovation in healthcare works to move the field from focusing on costly reactive care for established diseases to proactive preventive care, there is an emphasis on promoting self-care. New approaches from behavioral science have the potential for modifying individual oral health behaviors (Buunk-Werkhoven, Dijkstra, and Van Der Schans 2011). As a supplement to episodic clinician-delivered oral hygiene instruction, technologies can be leveraged to deliver engaging feedback and motivational messages to individuals around the time that they brush their teeth.

Towards that goal, our main objective is to build an online reinforcement learning (RL) algorithm (Sutton and Barto 2018) to be incorporated into the Oralytics mobile health application. Oralytics is a mobile health smartphone app that provides interventions to encourage oral self-care behaviors. The RL algorithm we develop will learn, online, users' responsivity to messages and decide at each of two brushing times per day whether to deliver each user an intervention message; the goal of the algorithm is to improve users' brushing quality throughout the study. The first Oralytics clinical trial with an RL algorithm to optimize message delivery will begin in early 2023.

The most commonly used RL algorithm in mobile health interventions are *bandit algorithms* (Tewari and Murphy 2017), one of the simplest types of RL algorithms. Bandit algorithms are commonly used because they can run reliably and stably in an online environment. Mobile health clinical trials can require years of work by an interdisciplinary team to develop. Changing the RL algorithm once the study has begun jeopardizes trial validity. So it is critical that the RL algorithm run stably (e.g., the algorithm updates quickly enough to select actions each day) (Trella et al. 2022).

However, classical bandit algorithms, designed to optimize immediate rewards, are not equipped to account for the delayed effects of actions (i.e., the impact of a current intervention on a user's responsiveness to future interventions). Inappropriate (e.g., untimely or too many) notification interruptions may cause user burden. Preventing user burden is crucial, especially in mobile health settings, which typically have high user dropout rates (Meyerowitz-Katz et al. 2020; Amagai et al. 2022). One option is to use RL algorithms that model a full Markov decision process (MDP), specifically by modeling how a given action impacts the next state and future rewards achievable when starting in that next state. However, because of the constrained setting of Oralytics,

specifically, the highly noisy outcomes and limited data per user, we do not expect to have enough data to learn the large number of parameters needed in a MDP-based algorithm. Moreover, MDP-based algorithms may not be as stable as bandit algorithms with respect to running and updating online. Bandit algorithms can also be interpreted as a form of discount regularization for a full MDP-based RL algorithm (Jiang et al. 2015).

Contributions: In this work, we build on the optimal reward problem (ORP) approach (Sorg 2011) and the theoretical work of (Jain et al. 2021) in model predictive control (MPC) to design a surrogate reward with a cost term that captures the delayed negative effects of sending messages (Section 3). Note that we will continue to *evaluate* our algorithms in terms of the original reward (true target), but will use surrogate rewards penalized by the cost term to optimize the *learning* of the algorithm. Additionally, we relate the cost term on the reward to a proxy for the impact of actions on the future state in the classical Bellman equation. Including such a cost term helps to capture the delayed effects of actions, allows us to continue to use bandit algorithms, and does not require the algorithm to learn additional parameters online.

To evaluate the design of the cost term, we first build realistic simulation environment test beds using previously collected user brushing data (Section 2.1). We then use domain expertise and simulation test beds to evaluate the design of the cost terms in their ability to capture the delayed negative effects of sending a message (Section 5). An interesting takeaway from this work is that *designing the rewards used by the algorithm is a critical component of real-world RL algorithm design.*

Supplement and Code: Throughout the paper we reference sections of the supplement that further discuss topics presented in this paper. The supplement can be found here: https://bit.ly/3PcU7fE and all code used in this paper can be found here: https://bit.ly/3JL21f5.

2 Preliminaries

The Oralytics RL algorithm will deliver feedback and motivational messages to users via notifications on a smartphone app. The first clinical trial of Oralytics with the RL algorithm will consist of approximately $N = 72$ users. Each user will participate in the study for 70 days. Each day there are two decision times, one hour before the user-specified morning and evening brushing times. This means for each user, there are a total of $T = 140$ decision times. At each decision time $t \in [1, 140]$ for user i, the algorithm decides between action $A_{i,t} = 1$ (send a message) and $A_{i,t} = 0$ (do not send a message) given the user's current state $S_{i,t}$. After action $A_{i,t}$ is executed in state $S_{i,t}$, a reward, $R_{i,t}$, is observed. Note that although the algorithm's action space is binary, if the algorithm does decide to send a message, there is a more complex procedure for selecting a specific message prompt. Please see Section 3.7 of the supplement for more details on message selection.

The goal of this paper is to design the reward for the Oralytics algorithm, which optimizes the desired health outcome while also preventing user burden, especially when needing to use a simple algorithm due to real-world constraints. The reward design will enhance the ability of the Oralytics algorithm to learn fast and select good actions stably in the real-world study. In the remainder of this section, we first discuss the available data from previous studies used to inform the design of the RL algorithm, and then we discuss the RL algorithm. In the following sections, we will discuss the design of the reward.

2.1 Available Data from Previous Studies

One of the main challenges is that we want to use available data to inform the design of the RL algorithm, but the available data is sparse and only partially informative. We have access to two data sets on brushing behaviors, ROBAS 2 (Shetty et al. 2020) and ROBAS 3. More importantly, both data sets only have observations under no actions.

ROBAS 3 has a more sophisticated sensory suite (the same suite that will be used in Oralytics) than ROBAS 2. In addition to brushing duration, ROBAS 3 collected sensory information (i.e., brushing pressure) to indicate brushing quality, whereas ROBAS 2 recorded brushing duration only. However, the study population of ROBAS 3 was slightly different because ROBAS 3's main goal was to test the robustness of the new passive data collection system. More importantly, ROBAS 3 had fewer users than ROBAS 2 and those users had worse brushing performance (i.e., lower average brushing duration and more sessions with no brushing). Table 1 highlights additional key differences between the ROBAS 2 and ROBAS 3 data sets.

We use the ROBAS 2 data set to design the prior used in the RL algorithm. We use the ROBAS 3 data set to construct a quality simulation environment that serves as a test bed for evaluating the RL algorithm candidates.

Property	ROBAS 2	ROBAS 3
Num. Users	32	13
Num. Datapoints	1792	2188
Intervention Messages	No	No
Sensor Data	No	Yes
Avg. Brushing Duration	85.874	78.098
% Sessions with No Brushing	38%	48%

Table 1: Differences between the ROBAS 2 and ROBAS 3 data sets

2.2 RL Algorithm

The main goal of many RL problems is to maximize accumulated rewards. To accomplish this goal, online RL algorithms (i) learn a model of the environment and (ii) have an action selection strategy. In Oralytics, we build on a contextual bandit framework which (i) learns a reward approximating function (model for the mean reward given the current state and action, $\mathbb{E}[R_{i,t}|S_{i,t}, A_{i,t}]$), and (ii) uses the learned model to select actions based on the current state $S_{i,t}$.

Pursuing a full MDP-based RL algorithm involves modeling both state transition probabilities and the expected reward for each state and action to form a policy. As discussed

in the introduction, due to the constrained setting of Oralytics, we do not expect to have enough data to effectively learn parameters for a full MDP-based algorithm. Thus, we modify a contextual bandit algorithm—specifically, we use a variant of linear posterior sampling (Russo et al. 2017). Posterior sampling involves using a Bayesian model for the mean reward and selection actions according to the posterior probability that each action is optimal. The Bayesian framework allows us to incorporate data from previous studies and domain knowledge into the prior distribution. Further, the actions are selected probabilistically, which facilitates after-study inference (Yao et al. 2021; Zhang, Janson, and Murphy 2020).

For our posterior sampling algorithm, we use Bayesian Linear Regression (BLR) with action centering for the reward approximating function. BLR with action centering has a closed-form posterior update, requires access only to treatment effect features at decision times, and is robust to miss-specification of the correctness of the baseline reward model (Liao et al. 2019). Specifically, the BLR with action centering posterior sampling algorithm uses the following model of the reward:

$$R_{i,t} = m(S_{i,t})^T \alpha_0 + \pi_{i,t} f(S_{i,t})^T \alpha_1 + (A_{i,t} - \pi_{i,t}) f(S_{i,t})^T \beta + \epsilon \quad (1)$$

where $\pi_{i,t}$ is the probability that the RL algorithm selects action $A_{i,t} = 1$ for user i in state $S_{i,t}$, and $\epsilon \sim \mathcal{N}(0, \sigma^2)$. We put a $\mathcal{N}(\mu_{\alpha_0}, \Sigma_{\alpha_0})$ prior on α_0, a $\mathcal{N}(\mu_{\alpha_1}, \Sigma_{\alpha_1})$ prior on α_1, and a $\mathcal{N}(\mu_\beta, \Sigma_\beta)$ prior on β. We discuss the procedure for setting prior values for $\sigma^2, \mu_{\alpha_0}, \Sigma_{\alpha_0}, \mu_\beta, \Sigma_\beta$ using ROBAS 2 data in Section 3.3 of the supplement.

Finally, we design the Oralytics RL algorithm to learn by pooling the data of all users in the study. Algorithms that learn using all users' data can learn faster than those that learn using a single user's data. Moreover, in early experiments, BLR with full pooling performed better than BLR that learns using only a single user's data or a smaller cluster of users' data (Trella et al. 2022). In Equation (1), the reward model parameters are not indexed by the user i to reflect how we are learning a single RL algorithm for all users in the study. For further details on the RL algorithm, such as how the algorithm performs action selection and posterior updating, please see Section 3 of the supplement.

3 Reward Design for the Oralytics Algorithm

We evaluate our Oralytics RL algorithm in terms of its ability to maximize each user's total brushing quality $\sum_{t=1}^{T} Q_{i,t}$, where $Q_{i,t}$ is a non-negative measure of brushing quality observed after each decision time (two times a day). In collaboration with domain scientists on our team, we chose $Q_{i,t} = \min(B_{i,t} - P_{i,t}, 180)$, where $B_{i,t}$ is the user's brushing duration in seconds, and $P_{i,t}$ is the aggregated duration of overpressure in seconds. $Q_{i,t}$ is truncated to be at most 180 to avoid optimizing for over-brushing. $B_{i,t}$ and $P_{i,t}$ capture brushing quality as healthy brushing behaviors consist of brushing the dentist-recommended 120 seconds and using less pressure when brushing. For further discussion of the definition of brushing quality, please see Section 2.1 of the supplement.

We now discuss the design of the reward that will be used by the RL algorithm. Throughout, we are interested in optimizing for brushing quality $Q_{i,t}$ and refer to $R_{i,t}$, the reward used by the RL algorithm, as the *surrogate reward*. The RL uses surrogate rewards $R_{i,t}$ to update its parameters. Let $R_{i,t} \in \mathbb{R}$ denote the surrogate reward for the ith user at decision time t:

$$R_{i,t} := Q_{i,t} - C_{i,t} \quad (2)$$

Conceptually, the cost term is designed to allow the RL algorithm to optimize for healthy brushing behavior while simultaneously considering the effect of the current message on the effectiveness of future interventions. Based on domain knowledge, we believe that sending a message at a decision time t can only have a non-negative effect on the user's immediate brushing, $Q_{i,t}$. However, sending too many messages can risk habituation or may burden the user, thus affecting user responsiveness to future messages (i.e., affecting $Q_{i,t+1}, Q_{i,t+2}, \ldots, Q_{i,T}$). Therefore, to anticipate these negative delayed effects of sending a message, we reduce the algorithm's reward when negative delayed effects are likely to occur. $C_{i,t}$ provides this reduction, as including $C_{i,t}$ in the algorithm's reward will provide a signal that sending a message ($A_{i,t} = 1$) may negatively affect future states. This signal is needed because we are using a contextual bandit algorithm that does not explicitly model the delayed effects of actions.

In fact, $C_{i,t}$ can be viewed as a crude proxy for the delayed effect of actions in the Bellman equation in a MDP environment. Recall that according to the Bellman equation, it is optimal to select action 1 over action 0 if the immediate expected reward received from action 1 over action 0 exceeds the difference in optimal "future values" of selecting action 0 over action 1; specifically action 1 and action 0 can differ in "future value" due to the probability that each action will lead to a favorable or less favorable next state. Mathematically, this difference in future value is $\mathbb{E}[V^*(S_{i,t+1})|S_{i,t}, A_{i,t} = 0] - \mathbb{E}[V^*(S_{i,t+1})|S_{i,t}, A_{i,t} = 1]$ where V^* is the optimal value function in a MDP setting. In a classical contextual bandit setting, this difference in future values of two actions is always zero. Including the cost term $C_{i,t}$ on selecting action 1 allows the contextual bandit algorithm to act in a more realistic setting in which there is some non-negative delayed effect of sending a message. For a full derivation, please see Section 3.6 in the supplement.

We define $\bar{B}_{i,t} := c_\gamma \sum_{j=1}^{14} \gamma^{j-1} Q_{i,t-j}$ and $\bar{A}_{i,t} := c_\gamma \sum_{j=1}^{14} \gamma^{j-1} A_{i,t-j}$. We set $\gamma = \frac{13}{14}$ to represent looking back 14 decision time points and scale each sum by constant $c_\gamma = \frac{1-\gamma}{1-\gamma^{14}}$ so that the weights sum to 1. Notice that our choice of γ and the scaling constant means $0 \leq \bar{B}_{i,t} \leq 180$ and $0 \leq \bar{A}_{i,t} \leq 1$. $\bar{B}_{i,t}$ captures the user's exponentially discounted brushing quality in the past week. $\bar{A}_{i,t}$ captures the number of interventions that were sent recently. Both terms are exponentially discounted because we expect that interventions sent and user brushing in the near past will be more predictive of the delayed impact of the actions (via burden

and/or habituation) than interventions sent and user brushing in the further past.

We define the cost of sending a message as:

$$C_{i,t} := \begin{cases} \xi_1 \mathbb{I}[\bar{B}_{i,t} > b]\mathbb{I}[\bar{A}_{i,t} > a_1] & \text{if } A_{i,t} = 1 \\ \quad +\xi_2 \mathbb{I}[\bar{A}_{i,t} > a_2] & \\ 0 & \text{if } A_{i,t} = 0 \end{cases} \quad (3)$$

Notice that the algorithm only incurs a cost if the current action is to send an intervention (i.e., $A_{i,t} = 1$). The first term $\xi_1 \mathbb{I}[\bar{B}_{i,t} > b]\mathbb{I}[\bar{A}_{i,t} > a_1]$ penalizes the reward if a high-performing user was sent too many messages within the past week. The second term $\xi_2 \mathbb{I}[\bar{A}_{i,t} > a_2]$ penalizes the reward, regardless of user performance, if a user was sent too many messages within the past week. b, a_1, a_2 are chosen by domain experts. Notice that $a_1 < a_2$ because we believe a high-performing user is more likely to be annoyed by a message. The scientific team decided to set the following values:

- $b = 111$ is set to the 50th percentile of user brushing durations in ROBAS 2, where brushing durations are truncated to 120 seconds if they exceed 120.
- $a_1 = 0.5$ represents a rough approximation of the user getting a message 50% of the time (rough approximation because we are using an exponential average mean)
- $a_2 = 0.8$ represents a rough approximation of the user getting a message 80% of the time (rough approximation because we are using an exponential average mean)

ξ_1, ξ_2 are non-negative hyperparameters that we tune in Section 5.

4 Related Work

4.1 Reward Design

Reward design is one of the foremost challenges in real-world reinforcement learning; it impacts the RL algorithm's ability to optimally select actions corresponding to the desired goal and the speed at which it learns (Mataric 1994). There are a variety of ways to modify and construct rewards. One example is reward shaping (Ng, Harada, and Russell 1999; Laud and DeJong 2003; Laud 2004). Here, one may design rewards to encourage RL algorithms to learn faster by encoding domain information (e.g., information about the relative effectiveness of different actions). Another example is intrinsically motivated RL (Chentanez, Barto, and Singh 2004), which designs the reward function to include the novelty of an experience. This design encourages exploration with the goal of helping the algorithm learn skills for different tasks. Additionally, in practice, rewards have been penalized to account for the cost of taking certain actions (Piette et al. 2022) and to satisfy domain-specific constraints (Tessler, Mankowitz, and Mannor 2018).

The idea here of designing a surrogate reward rather than using the true target is similar to the approach to the ORP (Sorg 2011). ORP distinguishes between the algorithm reward function (surrogate reward) that helps the algorithm learn and a separate target for the designer (true target) that is used to evaluate the algorithm. Optimal reward design should lead to a greater expected true target. A similar approach is optimizing a cost function in the MPC literature (Jain et al. 2021). Using MPC with the true cost function may not be the best choice due to errors in a dynamics model from the truncated planning horizon and the model approximations used. Their work shows that an MPC algorithm that uses a surrogate cost function (reward function) can have a lower true cost because it can mitigate problems that occur when the dynamics model is only approximate.

Similarly, in our setting, $\sum_{t=1}^{T} Q_{i,t}$ is the true target, but using $Q_{i,t}$ as the reward may not be the best choice in cases where the RL algorithm must be made simple due to real-world constraints (e.g., managing high noise by using a contextual bandit algorithm rather than a full MDP based RL algorithm and using an approximate model for the reward function). We penalize the reward with a cost term which, when used by the online RL algorithm, should lead to a higher value of the true target, $\sum_{t=1}^{T} Q_{i,t}$.

4.2 RL In Mobile Health

Aimed at behavioral change, many mobile health studies use contextual bandit algorithms to optimize interventions under real-world constraints as discussed in Trella et al. (2022) and Figueroa et al. (2021). Alternative approaches include Liao et al. (2019), which uses a modified version of a posterior sampling contextual bandit algorithm; Wang et al. (2021), which uses an MDP framework and learns an action selection strategy offline; and Zhou et al. (2018), which uses inverse RL to estimate a model of the reward and mixed-integer linear programming to select a 7-day schedule of personal step count goals for each user. Similar to this work, Liao et al. (2019) incorporated a term to approximate the delayed effects of the current action on future rewards. However, their approach did not involve explicitly developing a surrogate reward and they additionally did not allow the term modeling the delayed effects of actions to depend on previous user outcomes; it depended only on previous actions. Finally, they also did not develop simulation environments with delayed effects to evaluate the design of their delayed effects of actions nor did they provide explicit advice on how to choose the hyperparameters in their delayed effects term.

Previous work (Trella et al. 2022) discusses a preliminary design of the RL algorithm for Oralytics. However, the main contribution of that paper was offering a generalizable framework for designing and evaluating RL algorithms for digital interventions. Although we follow the framework discussed in Trella et al. (2022), this paper focuses on reward design, takes advantage of new data, specifically the ROBAS 3 dataset, and uses the collected ROBAS 2 dataset to form the prior in our RL algorithm.

5 Optimizing Hyperparameters in the Surrogate Reward

Recall in Equation (3), ξ_1, ξ_2 are hyperparameters that must be selected. We evaluate different values of ξ_1, ξ_2 in terms of their ability to maximize user brushing quality scores, $\sum_{t=1}^{T} Q_{i,t}$, across a variety of plausible environments. To do this, in each simulation environment variant, we perform

a grid search over the range of possible values of ξ_1, ξ_2. Specifically for each algorithm variant (different values of ξ_1, ξ_2) and simulation environment pair, we consider evaluating two criteria:

1. Average cumulative brushing quality across all users, $\frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T} Q_{i,t}$
2. 25th-percentile of cumulative brushing quality, $\sum_{t=1}^{T} Q_{i,t}$, across all users

We use these metrics to evaluate algorithm performance on both average and worse-off users.

5.1 Simulation Environment Design

Our simulation environment design involves three main components: (i) a base model of the environment under no intervention, (ii) initial treatment effect sizes for each user, and (iii) a procedure for modeling delayed effects of actions, specifically by shrinking users' treatment effect sizes.

Base Model of Environment Recall that we use the ROBAS 3 data set to construct the simulation environment. Even though ROBAS 3 did not involve intervention messages, we can still use the dataset to inform the base model for the simulation environment (i.e., a model for generating brushing quality under no action). For the base model, we fit a single zero-inflated Poisson model (Feng 2021) per user in the ROBAS 3 study. Each zero-inflated model's Bernoulli component represents a latent state of the user's intention to brush and the Poisson component represents brushing quality when the user intends to brush. We use a zero-inflated model because brushing quality is a highly zero-inflated outcome; many users miss brushing sessions altogether and have brushing quality zero (Trella et al. 2022).

Initial User Treatment Effects A user's responsiveness to an intervention is measured by their unique treatment effect sizes $\Delta_{i,B}$ and $\Delta_{i,N}$. $\Delta_{i,B}$ is the imputed treatment effect for the Bernoulli component and $\Delta_{i,N}$ is the imputed treatment effect for the Poisson component. The larger the effect sizes, the greater the user's responsiveness to the intervention. Since ROBAS 3 had no data under intervention, we impute $\Delta_{i,B}$ and $\Delta_{i,N}$ for each user by drawing a value from a zero-truncated normal distribution with mean and variance informed by the fitted parameters of the environment base model for that user $w_{i,b}$ and $w_{i,p}$. Further details on building the base simulation environment and imputing effect sizes can be found in Section 2 of the supplement.

Incorporating the base simulation model and the effect sizes, the environment generates brushing quality $Q_{i,t}$ under action $A_{i,t}$ in state $S_{i,t}$ using:

$$Z_{i,t} \sim \text{Bern}\left(1-\widetilde{p}_{i,t}\right)$$
$$\widetilde{p}_{i,t} = \text{sigmoid}\left(g(S_{i,t})^T w_{i,b} - A_{i,t}\max(h(S_{i,t})^T\Delta_{i,B}, 0)\right)$$
$$Y_{i,t} \sim \text{Pois}\left(\lambda_{i,t}\right)$$
$$\lambda_{i,t} = \exp\left(g(S_{i,t})^T w_{i,p} + A_{i,t}\max(h(S_{i,t})^T\Delta_{i,N}, 0)\right)$$
$$Q_{i,t} = Z_{i,t}Y_{i,t}$$

Above, $g(S_{i,t})$ is the baseline feature vector and $h(S_{i,t})$ is the feature vector that interacts with the effect size found in Sections 2.2.1 and 2.3.1 of the supplement.

Modeling Delayed Effects of Actions We model delayed effects of actions by shrinking users' responsiveness to interventions (i.e., their initial treatment effect sizes). We shrink users' effect sizes by a factor $E \in [0, 1)$ when a certain "habituation" criterion is met. The habituation criterion is if either of the two scenarios holds: (a) $\mathbb{I}[\bar{B}_{i,t} > b]$ (user brushes well) and $\mathbb{I}[\bar{A}_{i,t} > a_1]$ (user was sent too many messages for a healthy brusher), or (b) $\mathbb{I}[\bar{A}_{i,t} > a_2]$ (the user has been sent too many messages). The first time a user's habituation criterion has been met, the user's future effect sizes $\Delta_{i,B}, \Delta_{i,N}$ starting at time $t+1$ will be scaled down proportionally by E for some $E \in (0, 1)$ (the user is less responsive to treatment). Then after a week, at time $t+14$, we will check the habituation criterion again. If the habituation criterion is met again, the effect sizes will be further shrunken by a factor of E down to $E^2 \cdot \Delta_{i,B}, E^2 \cdot \Delta_{i,N}$ starting at time $t+15$. However, if the habituation criterion is not fulfilled, then the user recovers their original effect size $\Delta_{i,B}, \Delta_{i,N}$ starting at time $t+15$. This procedure continues until the user finishes the study. Notice that this means the user can only have their effect size shrunk at most once a week. This procedure simulates how the user may experience habituation (reduction in responsiveness to the messages), but after a week, if the RL algorithm does not intervene too much, the user may dis-habituate and recover their prior responsiveness.

5.2 Experiments and Results

We consider three values for shrinking the effect size, $E \in \{0, 0.5, 0.8\}$, for a total of three environment variants. For each environment and RL algorithm (determined by the ξ_1, ξ_2 candidate value pairing), we simulate a study with $N = 72$ users over $T = 140$ decision times. The N users are drawn with replacement from all the user environment models fitted using ROBAS 3 data. Since the Oralytics study is expected to recruit users into the study at a rate of about four users per week, we also simulate incremental recruitment by having 4 users enter the study per week. Following the procedure described above, for each environment variant, we generate two grids corresponding to the two evaluation criteria described at the beginning of this section (Section 5). Each square in a grid represents a criterion evaluated on a simulated study using values ξ_1, ξ_2 for the cost term, averaged across 100 Monte Carlo simulated trials. Figure 1 shows heat maps of evaluation criterion values for different values of ξ_1, ξ_2 across the three environment variants.

The primary takeaway from our simulation results from Figure 1 is that using a surrogate reward is beneficial. This is because, on all of the heatmaps in Figure 1 below, the worst performing RL algorithm is that which learned with reward parameters $\xi_1 = 0$ and $\xi_2 = 0$. This algorithm corresponding to learning with a zero cost term represents the classical bandit algorithm. Thus, even though there are different optimal values of ξ_1 and ξ_2 for each environment and each evaluation criterion, it appears that choosing *some* nonzero values of ξ_1 and ξ_2 is strictly beneficial.

Now consider the first evaluation criterion, the average cumulative brushing quality across all users, $\frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T} Q_{i,t}$. The left column of Figure 1 shows re-

sults with different levels of delayed effects of actions. Notice that as the strength of the delayed effects increases (i.e., E decreases) the penalty for burdening the users becomes harsher, and thus higher values of ξ_1 and ξ_2 are preferable. Note that for the $E = 0.8$ case, we believe $\xi_1 = 0$ is favorable because in this environment the delayed effects of actions are relatively weak and the penalty incurred by the delayed effects is not outweighed by the benefit of sending users additional messages (recall that ξ_1 weights the condition where the user brushes well and was sent a moderate amount of messages).

Regarding evaluation criterion 2, the 25th percentile of $\frac{1}{T}\sum_{t=1}^{T} Q_{i,t}$ across users, notice that the optimal value of cost term hyperparameter ξ_2 increases with the severity of the delayed effects (lower values of E means more severe delayed effects). The reason that $\xi_1 = 0$ is favored for this second evaluation criterion is that for these worse-off users, it was very rare for them

To select ξ_1, ξ_2, we want to balance providing high value for the average and the worst-performing user. Other factors that influence our decision include prioritizing environment variants that the scientific team considers most important and considering the variation of grid values in the heat maps. For example, we may be willing to forgo an optimal value in a grid with low variation (values perform similarly) than an optimal value in a grid with high variation. Thus, based on all our simulation results, we plan to choose intermediate values of both ξ_1 and ξ_2, specifically $\xi_1 = 100$ and $\xi_2 = 100$. The scientific team will reconsider these values as we collect more ROBAS 3 brushing data.

6 Discussion

In this paper, we describe the reward design for developing an online RL algorithm that will be deployed in Oralytics, a mobile health app promoting oral self-care behaviors. By designing a surrogate reward to include a cost term, instead of having the RL algorithm learn using the true target, we generalize the contextual bandit framework to deal with the key challenge of capturing negative, delayed effects of interventions. After evaluating these reward candidate values in our simulation test bed, we chose hyperparameter values of the reward that balances the performance of the RL algorithm for the average user and the worst-off user across all environment variants. As our development for Oralytics is ongoing, we will revisit the surrogate reward construction and will work with the scientific team in finalizing the algorithm that goes into the clinical trial.

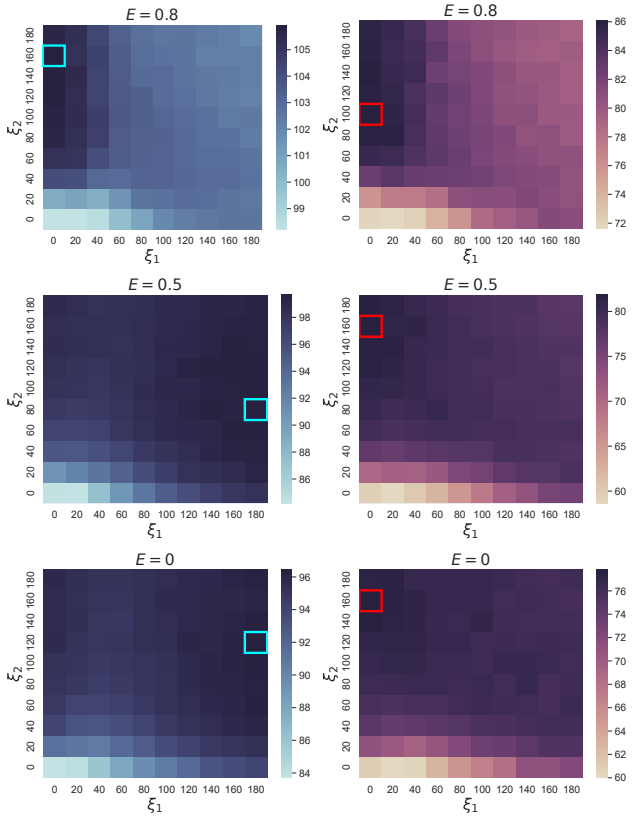

Figure 1: Heatmap of Candidate Values for ξ_1, ξ_2. We evaluate candidate values ξ_1, ξ_2 using two metrics and across three simulation environment variants. Effect size shrinkage E varies from slight shrinkage ($E = 0.8$) to severe shrinkage ($E = 0$) from top to bottom. The left column (blue) shows grids evaluated using average $\sum_{t=1}^{T} Q_{i,t}$ across users and the right column (purple) shows grids evaluated using 25th-percentile of $\sum_{t=1}^{T} Q_{i,t}$ across users. The grid with the highest criteria value is boxed for readability.

Acknowledgments

This research was funded by NIH grants IUG3DE028723, P50DA054039, P41EB028242, U01CA229437, UH3DE028723, and R01MH123804. KWZ is supported by the National Science Foundation grant number NSF CBET–2112085 and by the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE1745303. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. KWZ is also supported by the Siebel Scholars program class 2023.

References

Amagai, S.; Pila, S.; Kaat, A. J.; Nowinski, C. J.; Gershon, R. C.; et al. 2022. Challenges in participant engagement and retention using mobile health apps: literature review. *Journal of medical Internet research*, 24(4): e35120.

Benjamin, R. M. 2010. Oral health: the silent epidemic. *Public health reports*, 125(2): 158–159.

Buunk-Werkhoven, Y. A.; Dijkstra, A.; and Van Der Schans, C. P. 2011. Determinants of oral hygiene behavior: a study based on the theory of planned behavior. *Community dentistry and oral epidemiology*, 39(3): 250–259.

Chentanez, N.; Barto, A.; and Singh, S. 2004. Intrinsically motivated reinforcement learning. *Advances in neural information processing systems*, 17.

Feng, C. X. 2021. A comparison of zero-inflated and hurdle models for modeling zero-inflated count data. *Journal of Statistical Distributions and Applications*, 8(1): 1–19.

Figueroa, C. A.; Aguilera, A.; Chakraborty, B.; Modiri, A.; Aggarwal, J.; Deliu, N.; Sarkar, U.; Jay Williams, J.; and Lyles, C. R. 2021. Adaptive learning algorithms to optimize mobile applications for behavioral health: guidelines for design decisions. *Journal of the American Medical Informatics Association*, 28(6): 1225–1234.

Jain, A.; Chan, L.; Brown, D. S.; and Dragan, A. D. 2021. Optimal Cost Design for Model Predictive Control. In *Learning for Dynamics and Control*, 1205–1217. PMLR.

Jiang, N.; Kulesza, A.; Singh, S.; and Lewis, R. 2015. The dependence of effective planning horizon on model accuracy. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*. Citeseer.

Laud, A.; and DeJong, G. 2003. The influence of reward on the speed of reinforcement learning: An analysis of shaping. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, 440–447.

Laud, A. D. 2004. *Theory and application of reward shaping in reinforcement learning*. University of Illinois at Urbana-Champaign.

Liao, P.; Greenewald, K. H.; Klasnja, P. V.; and Murphy, S. A. 2019. Personalized HeartSteps: A Reinforcement Learning Algorithm for Optimizing Physical Activity. *CoRR*, abs/1909.03539.

Löe, H. 2000. Oral hygiene in the prevention of caries and periodontal disease. *International dental journal*, 50(3): 129–139.

Martin, L. R.; Williams, S. L.; Haskard, K. B.; and DiMatteo, M. R. 2005. The challenge of patient adherence. *Therapeutics and clinical risk management*, 1(3): 189.

Mataric, M. J. 1994. Reward functions for accelerated learning. In *Machine learning proceedings 1994*. Elsevier.

Meyerowitz-Katz, G.; Ravi, S.; Arnolda, L.; Feng, X.; Maberly, G.; Astell-Burt, T.; et al. 2020. Rates of attrition and dropout in app-based interventions for chronic disease: systematic review and meta-analysis. *Journal of Medical Internet Research*, 22(9): e20283.

Ng, A. Y.; Harada, D.; and Russell, S. 1999. Policy invariance under reward transformations: Theory and application to reward shaping. In *Icml*, volume 99, 278–287.

Piette, J. D.; Newman, S.; Krein, S. L.; Marinec, N.; Chen, J.; Williams, D. A.; Edmond, S. N.; Driscoll, M.; LaChappelle, K. M.; Maly, M.; et al. 2022. Artificial Intelligence (AI) to improve chronic pain care: Evidence of AI learning. *Intelligence-Based Medicine*, 100064.

Russo, D.; Roy, B. V.; Kazerouni, A.; and Osband, I. 2017. A Tutorial on Thompson Sampling. *CoRR*, abs/1707.02038.

Shetty, V.; Morrison, D.; Belin, T.; Hnat, T.; and Kumar, S. 2020. A Scalable System for Passively Monitoring Oral Health Behaviors Using Electronic Toothbrushes in the Home Setting: Development and Feasibility Study. *JMIR Mhealth Uhealth*, 8(6): e17347.

Sorg, J. D. 2011. *The optimal reward problem: Designing effective reward for bounded agents*. Ph.D. thesis, University of Michigan.

Sutton, R. S.; and Barto, A. G. 2018. *Reinforcement learning: An introduction*. MIT press.

Tessler, C.; Mankowitz, D. J.; and Mannor, S. 2018. Reward constrained policy optimization. *arXiv preprint arXiv:1805.11074*.

Tewari, A.; and Murphy, S. A. 2017. From ads to interventions: Contextual bandits in mobile health. In *Mobile Health*, 495–517. Springer.

Trella, A. L.; Zhang, K. W.; Nahum-Shani, I.; Shetty, V.; Doshi-Velez, F.; and Murphy, S. A. 2022. Designing Reinforcement Learning Algorithms for Digital Interventions: Pre-Implementation Guidelines. *Algorithms*, 15(8).

Wang, S.; Sporrel, K.; van Hoof, H.; Simons, M.; de Boer, R. D.; Ettema, D.; Nibbeling, N.; Deutekom, M.; and Kröse, B. 2021. Reinforcement learning to send reminders at right moments in smartphone exercise application: A feasibility study. *International Journal of Environmental Research and Public Health*, 18(11): 6059.

Yao, J.; Brunskill, E.; Pan, W.; Murphy, S.; and Doshi-Velez, F. 2021. Power Constrained Bandits. In Jung, K.; Yeung, S.; Sendak, M.; Sjoding, M.; and Ranganath, R., eds., *Proceedings of the 6th Machine Learning for Healthcare Conference*, volume 149 of *Proceedings of Machine Learning Research*, 209–259. PMLR.

Zhang, K.; Janson, L.; and Murphy, S. 2020. Inference for batched bandits. *Advances in neural information processing systems*, 33: 9818–9829.

Zhou, M.; Mintz, Y.; Fukuoka, Y.; Goldberg, K.; Flowers, E.; Kaminsky, P.; Castillejo, A.; and Aswani, A. 2018. Personalizing mobile fitness apps using reinforcement learning. In *CEUR workshop proceedings*, volume 2068. NIH Public Access.