Supplementary Material for "Reward Design For An Online Reinforcement Learning Algorithm For Supporting Oral Self-Care"

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

August 13, 2022

1 Data and Code

We use two data sets from previous studies: ROBAS 2 [Shetty et al., 2020] and ROBAS 3. The data sets are publically available here for ROBAS 2 and here for ROBAS 3. Code for this paper can be found in a github repository here.

2 Simulation Environments

2.1 Defining the Target: Brushing Quality $Q_{i,t}$

Brushing quality is the true target that the scientific team cares about in achieving healthy brushing behaviors (desired behavioral health outcome). We define a proxy for brushing quality as $Q_{i,t} = \min(B_{i,t} - P_{i,t}, 180)$. $B_{i,t}$ is the participant's brushing duration in seconds and $P_{i,t}$ is the aggregated duration of over pressure in seconds. $Q_{i,t}$ is truncated to avoid optimizing for over-brushing.

Other sensory information that were considered was zoned brushing duration, as another indicator of brushing quality is brushing more evenly across the six zones. However, zoned brushing duration is not reliably obtainable as it requires Bluetooth connectivity and the participant to stand close enough to the docking station. Therefore, zoned brushing duration only appeared in about 82% of brushing sessions in the pilot study. $Q_{i,t}$ can be interpreted as a non-penalized reward, and the true target capturing the health outcome that the scientific team's wants to maximize.

2.2 Building the Environment Base Model

2.2.1 Baseline Feature Space of the Environment Base Models

We form the following features using domain expert knowledge from behavioral health and dentistry. In this section we discuss how we fit a model for the

baseline reward (i.e., the brushing quality under action $A_{i,t} = 0$). A discussion on how we model brushing quality under action $A_{i,t} = 1$ is in Section 2.3.

$g(S_{i,t}) \in \mathbb{R}^6$ denotes the feature space used to fit a model of the baseline reward which is the following:

1. Bias / Intercept Term $\in \mathbb{R}$
2. Time of Day (Morning/Evening) $\in \{0, 1\}$
3. Prior Day Total Brushing Quality (Normalized) $\in \mathbb{R}$
4. Weekend Indicator (Weekday/Weekend) $\in \{0, 1\}$
5. Proportion of Non-zero Brushing Sessions Over Past 7 Days $\in [0, 1]$
6. Day in Study (Normalized) $\in [-1, 1]$

2.2.2 Normalization of State Features

We normalize features to ensure that all state features are within a similar range. The Prior Day Total Brushing Quality feature is normalized using z-score normalization (subtract mean and divide by standard deviation). Since each participant in the ROBAS 3 study had varying participant study lengths (due to dropping out), the Day in Study feature (originally in the range $[1 : T_i]$ where T_i represents the number of days user i was in the study) is normalized to be between $[-1, 1]$. Note that when generating rewards, Day in Study is normalized based on Oralytic's anticipated 70 day study duration (range is still $[-1, 1]$).

$$\text{Normalized Total Brushing Quality in Seconds} = (\text{Brushing Quality} - 154)/163$$

$$\text{Normalized Day in Study When Fitting Model for User } i = \left(\text{Day} - \frac{T_i + 1}{2}\right) / \frac{T_i - 1}{2}$$

$$\text{Normalized Day in Study When Generating Rewards} = (\text{Day} - 35.5)/34.5$$

2.2.3 Zero-Inflated Poisson Model for Brushing Quality

Due to the zero-inflated nature of the ROBAS 3 data set, we use a zero-inflated Poisson model to generate brushing quality. $w_{i,b}, w_{i,p}$ are user-specific weight vectors, $g(S_{i,t})$ is the baseline feature vector of the current state defined in Section 2.2.1, and $\text{sigmoid}(x) = \frac{1}{1+e^{-x}}$ is the sigmoid function.

To generate brushing quality for user i at decision time t, we use the following model:

$$Z \sim \text{Bernoulli}\left(1 - \text{sigmoid}(g(S_{i,t})^T w_{i,b})\right)$$

$$Y \sim \text{Poisson}\left(\exp\left(g(S_{i,t})^T w_{i,p}\right)\right)$$

$$\text{Brushing Quality} : Q_{i,t} = ZY$$

2.2.4 Fitting the Environment Base Model

We use ROBAS 3 data to fit the brushing quality model under no intervention ($A_{i,t} = 0$). We fit one model per user and each user model was fit using MAP with a prior $w_{i,b}, w_{i,p} \sim \mathcal{N}(0, I)$ as a form of regularization because we have sparse data for each user. We jointly fit parameters for both the Bernoulli and the Poisson components. Weights were chosen by running random restarts and selecting the weights with the highest log posterior density.

2.3 Imputing Treatment Effect Sizes for Simulation Environments

Recall that the ROBAS 3 data set does not have data under interventions (sending a message). Therefore we impute user-specific treatment effect sizes to model the reward under action 1. We use two guidelines to design the effect sizes:

1. For mobile health digital interventions, we expect the treatment effect (magnitude of weight) of actions to be smaller than (or on the order of) the effect of baseline features (baseline features are specified in Section 2.2.1).
2. The variance in treatment effects across users should be on the order of the variance in the effect of features across users (i.e., variance in parameters of fitted user-specific models).

We first construct a population-level effect size and then use the population effect size to sample unique effect sizes for each user. Following guideline 1 above, to set the population level effect size, we first take the absolute value of the weights (excluding that for the intercept term) of the user base models fitted using ROBAS 3 data and then average across users and features (e.g., the average absolute value of weight for time of day). To generate user-specific effect sizes, for each user, we draw a value from a truncated normal centered at the population effect sizes. Following guideline 2, the variance of the truncated normal distributions is found by again taking the absolute value of the weights of the base models fitted for each user, averaging the weights across features, and taking the empirical variance across users. A more detailed procedure for imputing effect sizes is found in Section 2.3.3.

2.3.1 Treatment Effect Feature Space

The treatment effect (advantage) feature space was made after discussion with domain experts on which features are most likely to interact with the intervention (action). $h(S_{i,t}) \in \mathbb{R}^5$ denotes the features space used for the treatment effect which is the following:

1. Bias/Intercept Term $\in \mathbb{R}$
2. Time of Day (Morning/Evening) $\in \{0, 1\}$
3. Prior Day Total Brushing Quality (Normalized) $\in \mathbb{R}$

4. Weekend Indicator (Weekday/Weekend) $\in \{0, 1\}$

5. Day in Study (Normalized) $\in \mathbb{R}$

2.3.2 Environment Model Including Effect Sizes

We impute treatment effects on both the Bernoulli component which represents the user's intent to brush and the Poisson component which represents the user's brushing quality when they intend to brush. After incorporating treatment effects, brushing quality $Q_{i,t}$ under action $A_{i,t}$ in state $S_{i,t}$ is:

$$Z \sim \text{Bernoulli}(1 - \text{sigmoid}(g(S_{i,t})^T w_{i,b} - A_{i,t} * \max(h(S_{i,t})^T \Delta_{i,B}, 0)))$$

$$Y \sim \text{Poisson}(\exp(g(S_{i,t})^T w_{i,p} + A_{i,t} * \max(h(S_{i,t})^T \Delta_{i,N}, 0)))$$

$$Q_{i,t} = ZY$$

$\Delta_{i,B}, \Delta_{i,N}$ are user-specific effect sizes. $g(S_{i,t})$ is the baseline feature vector as described in Section 2.2.1, and $h(S_{i,t})$ is the feature vector that interacts with the effect size specified above.

2.3.3 Procedure For Imputing Effect Sizes

We consider a unique realistic effect size for each user. We first construct a population level effect size for the Bernoulli and Poisson components, Δ_B, Δ_N respectively. We then use Δ_B, Δ_N to sample user-specific effect sizes $\Delta_{i,B}, \Delta_{i,N}$.

Recall that for the environment base model, we fit a user-specific model for the brushing quality and obtained user-specific parameters $w_{i,b}, w_{i,p} \in \mathbb{R}^6$ (values of the fitted parameters can be found here). Therefore we use the fitted parameters to form the population effect sizes as follows:

- $\Delta_B = \mu_{B,\text{avg}}$ where $\mu_{B,\text{avg}} = \frac{1}{5} \sum_{d \in [2:\,6]} \frac{1}{N} \sum_{i=1}^{N} |w_{i,b}^{(d)}|$.
- $\Delta_N = \mu_{N,\text{avg}}$ where $\mu_{N,\text{avg}} = \frac{1}{5} \sum_{d \in [2:\,6]} \frac{1}{N} \sum_{i=1}^{N} |w_{i,p}^{(d)}|$.

We use $w_{i,b}^{(d)}, w_{i,p}^{(d)}$ to denote the d^{th} dimension of the vector $w_{i,b}, w_{i,p}$ respectively; we take the average over all dimensions excluding $d = 1$, which represents the weight for the bias/intercept term.

Now to construct the user-specific effect sizes, we draw effect sizes for each each user from truncated normal distributions:

$$\Delta_{i,B} \sim \text{Truncated Normal}(\Delta_B, \sigma_B^2)$$

$$\Delta_{i,N} \sim \text{Truncated Normal}(\Delta_N, \sigma_N^2)$$

Notice that our design means the effect size on the Bernoulli component must be negative and the effect size on the Poisson component must be positive. If this is not the case, then that means in the current context, not sending a

message will yield a higher brushing quality than sending a message. This is unreasonable because the only negative consequences of sending a message is the diminishing the responsivity to future messages. We first truncate the sampled effect size at 0 and also ensure the output of the context dotted with the effect size will also be non-negative to prevent the effect size from switching signs and having a negative effect on brushing quality.

σ_B^2, σ_N^2 are the empirical standard deviation (SD) over the average of the fitted parameters and are set by the following procedure:

- σ_B is the empirical SD over $\{\mu_{i,B}\}_{i=1}^N$ where $\mu_{i,B} = \frac{1}{5}\sum_{d\in[2:\,6]} |w_{i,b}^{(d)}|$.
- σ_N is the empirical SD over $\{\mu_{i,N}\}_{i=1}^N$ where $\mu_{i,N} = \frac{1}{5}\sum_{d\in[2:\,6]} |w_{i,p}^{(d)}|$.

After the procedure described above, we found $\Delta_B = 0.743, \Delta_N = 0.227$, $\sigma_B = 0.177$, and $\sigma_N = 0.109$ (values are rounded to the nearest 3 decimal places).

2.4 Simulating Delayed Effects

To simulate delayed effects, we consider a user's responsiveness to an intervention, measured by the magnitude of that user's $\Delta_{i,B}, \Delta_{i,N}$. We define unresponsiveness as shrinking the user's environment effect size by some scalar E where $0 < E < 1$ (E varies between environment variants). At the first time t, if $\mathbb{I}[\bar{B}_{i,t} > b]$ (user brushes well) but $\mathbb{I}[\bar{A}_{i,t} > a_1]$ (user was sent too many messages) or if $\mathbb{I}[\bar{A}_{i,t} > a_2]$, then we move the user into the unresponsive state and starting at time $t+1$, the effect sizes of the user environment will be $E\cdot\Delta_{i,B}, E\cdot\Delta_{i,N}$. Then after a week, at time $t+14$, we will check the condition again. If $\mathbb{I}[\bar{B}_{i,t} > b]$ but $\mathbb{I}[\bar{A}_{i,t} > a_1]$ or if $\mathbb{I}[\bar{A}_{i,t} > a_2]$, the effect sizes will be $E^2\cdot\Delta_{i,B}, E^2\cdot\Delta_{i,N}$ starting at time $t+15$. However, if the condition 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.

2.4.1 Defining User-Specific Probability of Being Unresponsive

We consider a user-specific probability of being unresponsive π_{u}^i from a Beta distribution with mean equal to the population probability of being unresponsive $\mu_u = 0.5$, and variance $\sigma_u^2 = \left(\frac{\sigma_B+\sigma_N}{2}\right)^2$. To construct each π_{u}^i:

$$\pi_u^i \sim \text{Beta}(a,b), \quad \text{where} \quad a = \left(\frac{1-\mu_u}{\sigma_u^2} - \frac{1}{\mu_u}\right)\mu_u, \quad b = a\left(\frac{1}{\mu_u} - 1\right) \tag{1}$$

3 RL Algorithm

For the RL algorithm, we use a contextual bandit algorithm with Thompson sampling, a Bayesian Linear Regression reward function, and full pooling.

3.1 Feature Space of the RL Algorithm

$S_{i,t} \in \mathbb{R}^d$ represents the ith participant's state at decision time t, where d is the number of features describing the participant's state.

Advantage Feature Space $f(S_{i,t}) \in \mathbb{R}^4$ denotes the feature space used to predict the advantage (i.e. the immediate treatment effect) which is the following:

1. Bias / Intercept Term $\in \mathbb{R}$
2. Time of Day (Morning/Evening) $\in \{0, 1\}$
3. Exponential Average of Brushing Over Past 7 Days (Normalized) $\in \mathbb{R}$
4. Exponential Average of Messages Sent Over Past 7 Days $\in [0, 1]$

The normalization procedure for Prior Day Brushing Quality is the same as the one described in Section 2.2.2. Features 3 and 4 are $\bar{B} = c_\gamma \sum_{j=1}^{14} \gamma^{j-1} B_{i,t-j}$ and $\bar{A} = c_\gamma \sum_{j=1}^{14} \gamma^{j-1} A_{i,t-j}$ respectively, the same $\bar{B}, \bar{A}$ used in the cost term of the reward.

Baseline Feature Space $m(S_{i,t}) \in \mathbb{R}^5$ denotes the feature space used to predict the baseline reward function which contains all the above covariates and the following:

5. Weekend Indicator (Weekday/Weekend) $\in \{0, 1\}$

The feature space used by the RL algorithm candidates is different than the feature space used to model the reward in the simulation environments in order to test the robustness of the RL algorithm despite having a misspecified reward model.

3.2 Bayesian Linear Regression with Action Centering

Recall that our model for the reward is a Bayesian Linear Regression (BLR) model with action centering. Since we consider an algorithm that does full pooling (clustering with cluster size $K = N$), the algorithm is shared between all users in the study and therefore shares parameters $\alpha_0, \alpha_1, \beta$.

$$R_{i,t} = m(S_{i,t})^T \alpha_0 + \pi_{i,t} f(S_{i,t})^T \alpha_1 + (A_{i,t} - \pi_{i,t}) f(S_{i,t})^T \beta + \epsilon \tag{2}$$

where $\pi_{i,t}$ is the probability that the RL algorithm selects action $A_{i,t} = 1$ in state $S_{i,t}$ for participant i at decision time t. $\epsilon \sim \mathcal{N}(0, \sigma^2)$ and there are priors on $\alpha_0 \sim \mathcal{N}(\mu_{\alpha_0}, \Sigma_{\alpha_0})$, $\alpha_1 \sim \mathcal{N}(\mu_\beta, \Sigma_\beta)$, $\beta \sim \mathcal{N}(\mu_\beta, \Sigma_\beta)$. We discuss how we set informative prior values for $\mu_{\alpha_0}, \Sigma_{\alpha_0}, \mu_\beta, \Sigma_\beta, \sigma^2$ using the ROBAS 2 data set in Section 3.3.

3.3 Fitting the Prior for the Reward Function

We use the ROBAS 2 dataset to inform priors on $\alpha_0, \alpha_1, \beta$ as well as setting the noise variance term σ^2 in Equation 2. We follow the procedure as mentioned in [Liao et al., 2019]. Values are shown in Table 1.

Parameter	Value
σ^2	3396.449
μ_{α_0}	$[0, 4.925, 0, 0, 82.209]^T$
Σ_{α_0}	$\text{diag}(29.090^2, 30.186^2, 29.624^2, 12.989^2, 46.240^2)$
μ_β	$[0, 0, 0, 0]^T$
Σ_β	$29.624^2 \cdot I_4$

Table 1: **Prior values for the RL algorithm informed using the ROBAS 2 data set.** Values are rounded to the nearest 3 decimal places. After performing Generalized Estimating Equations' (GEE) analysis, we found the Prior Day Total Brushing Duration feature and the bias term to be significant.

3.3.1 Fitting σ^2

σ^2 is set using ROBAS 2 and fixed for the entire study. To choose the value, we fit a separate Generalized Estimating Equations' (GEE) linear regression model per user and set σ^2 to be the average of the empirical variance of the residuals for each user model across the 32 user models.

3.3.2 Fitting $\mu_{\alpha_0}, \Sigma_{\alpha_0}, \mu_\beta, \Sigma_\beta$

To set values for $\mu_{\alpha_0}, \Sigma_{\alpha_0}$, we follow the procedure described in [Liao et al., 2019]. We do the following: 1) conduct GEE regression analyses using all participant's data in ROBAS 2 in order to assess the significance of each feature; 2a) for features that are significant in the GEE analysis, the prior mean is set to be the point estimate found from GEE analysis and the prior standard deviation is set to be the standard deviation across user models from GEE analysis; 2b) for features that are not significant, the prior mean is 0 and we shrink the standard deviation by half. For feature 3. Exponential Average of Brushing Over Past 7 Days, we imputed that value with the average of past brushing obtained so far for the first week. Recall that ROBAS 2 had no data under $A_{i,t} = 1$, so for feature 4. Exponential Average of Messages Sent Over Past 7 Days, we set the prior mean to 0 and set the standard deviation to the average prior SD of the other features.

Notice that Σ_{α_0} is a diagonal matrix where the diagonal values are the prior variances described above. Notice that ROBAS 3 had a more sophisticated sensory suite than ROBAS 2, so while the scientific team chose brushing quality as the reward for the RL algorithm, we only have brushing duration from ROBAS 2. However, brushing duration is a reasonable guess for brushing quality, espe-

cially because average pressure duration is small. Therefore, when performing GEE analysis, the target is brushing duration.

Again, because ROBAS 2 only had no data under $A_{i,t} = 1$ so we cannot use the same procedure as described above to set μ_β, Σ_β. Instead, we set $\mu_\beta = \mathbf{0}$ and set $\Sigma_\beta = \sigma_\beta^2 I$ (diagonal matrix where each entry on the diagonal is σ_β^2). σ_β is equal to the average prior SD of the other features discussed above.

3.4 Posterior Updating

During the update step, the reward approximating function will update the posterior with newly collected data. Since we chose a full pooling algorithm, the algorithm will update the posterior using data shared between all users in the study. Additionally, we make M draws of the parameters from the updated posterior and use them for all decision times until the next update time. Here are the procedures for how the Bayesian linear regression model performs posterior updating.

Suppose we are selecting actions for decision time t. Let $\phi(S_{i,t}, A_{i,t}) = [m(S_{i,t}),\ \pi_{i,t} f(S_{i,t}),\ (A_{i,t} - \pi_{i,t}) f(S_{i,t})]^T$ be the joint feature vector and $\theta = [\alpha_0, \alpha_1, \beta]$ be the joint weight vector. Notice that Equation 2 can be vectorized in the form: $R_{i,t} = \phi(S_{i,t}, A_{i,t})\theta + \epsilon$. Notice that θ is shared amongst users because we are performing full pooling. Let $\Phi_{1:j-1} \in \mathbb{R}^{Kx5+4+4}$ be the matrix of all stacked vectors $\{\phi(S_{i,t}, A_{i,t})\}$, a matrix of all users' data that have been collected in the study up to time-step j, and $\mathbf{R}_{1:j-1} \in \mathbb{R}^K$ be a vector of stacked rewards $\{R_{i,t}\}$, a vector of all users' rewards that have been collected in the study up to time-step j.

Recall that we have normal priors on θ_i where $\theta_i \sim \mathcal{N}(\mu_{\text{prior}}, \Sigma_{\text{prior}})$, where $\mu_{\text{prior}} = [\mu_{\alpha_0}, \mu_\beta, \mu_\beta] \in \mathbb{R}^{4+4+5}$ and $\Sigma_{\text{prior}} = \text{diag}(\Sigma_{\alpha_0}, \Sigma_\beta, \Sigma_\beta)$. At current time j, the posterior distribution of the weights given current history H_{j-1}, $p(\theta|H_{j-1})$ is conjugate and is also normal.

$$\theta|H_{j-1} \sim \mathcal{N}(\mu_{j-1}^{post}, \Sigma_{j-1}^{post})$$

$$\Sigma_{j-1}^{post} = \left(\frac{1}{\sigma^2}\Phi_{1:j-1}^T \Phi_{1:j-1} + \Sigma_{prior}^{-1}\right)^{-1}$$

$$\mu_{j-1}^{post} = \Sigma_{j-1}^{post}\left(\frac{1}{\sigma^2}\Phi_{1:j-1}^T \mathbf{R}_{1:j-1} + \Sigma_{prior}^{-1}\mu_{prior}\right)$$

3.5 Action Selection

Our action selection scheme at decision time selects action $A_{i,t} \sim \text{Bern}(\pi_{i,t})$ where $\pi_{i,t} = \text{clip}\left(\tilde{\pi}_{i,t}\right)$. clip is the clipping function defined in Equation 3 and $\tilde{\pi}_{i,t}$ is the posterior probability that $A_{i,t} = 1$ is optimally defined in Section 3.5.1.

3.5.1 Posterior Sampling

Based on the Bayesian linear regression model of the reward, specified by Equation (2):

$$\tilde{\pi}_{i,t} = \Pr_{\tilde{\beta} \sim \mathcal{N}(\mu_{t-1}^{post}, \Sigma_{t-1}^{post})} \left\{ f(S_{i,t})^T \tilde{\beta} > 0 \middle| S_{i,t}, H_{t-1} \right\}$$

Note that the randomness in the probability above is only over the draw of $\tilde{\beta}$ from the posterior distribution.

3.5.2 Clipping to Form Action Selection Probabilities

Since we want to facilitate after-study analyses, we clip action selection probabilities using the action clipping function for some $\pi_{\min}, \pi_{\max}$ where $0 < \pi_{\min} \leq \pi_{\max} < 1$ is chosen by the scientific team:

$$\text{clip}(\pi) = \min(\pi_{\max}, \max(\pi, \pi_{\min})) \in [\pi_{\min}, \pi_{\max}] \tag{3}$$

For our simulations, $\pi_{\min} = 0.1, \pi_{\max} = 0.9$.

3.6 Relating the Cost Term to the Bellman Equation

Below is the Bellman Equation [Sutton and Barto, 2018] of the value $Q^\pi(s, a)$ (immediate and future value) of taking action a in state s, under policy π:

$$\begin{aligned} Q^\pi(s,a) &= \mathbb{E}_\pi \left[\sum_{t=1}^{T} \gamma^{t-1} R_{i,t} \middle| S_{i,t} = s, A_{i,t} = a \right] \\ &= \mathbb{E}[R_{i,t} | S_{i,t} = s, A_{i,t} = a] + \gamma \sum_{s' \in \mathcal{S}} p(S = s' | s, a) Q^\pi(s', a) \end{aligned} \tag{4}$$

The algorithm performs posterior sampling by selecting action $A_{i,t} = 1$ with probability:

$$\begin{aligned} \pi_{i,t} &= P\big[Q^\pi(s, a=1) - Q^\pi(s, a=0) > 0 \big| \text{Data}\big] \\ &= P\big[\mathbb{E}[R_{i,t} | S_{i,t} = s, A_{i,t} = 1] - \mathbb{E}[R_{i,t} | S_{i,t} = s, A_{i,t} = 0] > \eta \big| \text{Data}\big] \end{aligned} \tag{5}$$

In a full RL setting, $\eta = \gamma \mathbb{E}[V^*(s') | S_{i,t} = s, A_{i,t} = 0] - \gamma \mathbb{E}[V^*(s') | S_{i,t} = s, A_{i,t} = 1)$, where V^* is the optimal value function. In a bandit setting, $\eta = 0$.

Suppose $r_\theta(s, a)$ is the model we use for the mean reward $\mathbb{E}[R_{i,t} | S_{i,t} = s, A_{i,t} = a]$, where θ is the parameter of the model drawn from the posterior $p(\theta | D)$. In a bandit framework with posterior sampling, Equation 5 becomes:

$$\pi_{i,t} = P[r_\theta(s, a=1) - r_\theta(s, a=0) > 0; \theta \sim p(\theta | D])] \tag{6}$$

Using the surrogate reward designed in the paper, $R_{i,t} = Q_{i,t} - C_{i,t}$,

$$\implies r_\theta(s, a=1) - r_\theta(s, a=0) = \mathbb{E}[Q_{i,t} - C_{i,t} | S_{i,t} = s, A_{i,t} = 1, \theta]$$

$$-\mathbb{E}[Q_{i,t}|S_{i,t}=s,A_{i,t}=0,\theta]$$

$$=\mathbb{E}[Q_{i,t}|S_{i,t}=s,A_{i,t}=1,\theta]-\mathbb{E}[Q_{i,t}|S_{i,t}=s,A_{i,t}=0,\theta]-C_{i,t}$$

$$\implies \pi_{i,t}=P[\mathbb{E}[Q_{i,t}|S_{i,t}=s,A_{i,t}=1,\theta]$$
$$-\mathbb{E}[Q_{i,t}|S_{i,t}=s,A_{i,t}=0,\theta]>C_{i,t};\theta\sim p(\theta|D])]$$

The equation above shows a helpful interpretation that we are performing action selection at decision time with a threshold value $\eta = C_{i,t}$. Notice that the posterior distribution of parameters θ will be updated using the surrogate rewards, while traditional threshold approaches updates parameters using the true target $Q_{i,t}$. Finally, $C_{i,t}$ can be seen as an approximation for $\gamma\mathbb{E}[V^*(s')|S_{i,t}=s,A_{i,t}=0]-\gamma\mathbb{E}[V^*(s')|S_{i,t}=s,A_{i,t}=1)$. Therefore adding the cost term helps us capture the delayed effects of actions in a simple way, and allows us to continue to use bandit algorithms in settings where full RL is infeasible.

References

[Liao et al., 2019] 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.

[Shetty et al., 2020] 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.

[Sutton and Barto, 2018] Sutton, R. S. and Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.