Intelligent Pooling in Thompson Sampling for Rapid Personalization in Mobile Health

Sabina Tomkins [*1] **Peng Liao** [*1 2] **Serena Yeung** [1 3] **Predrag Klasnja** [2] **Susan Murphy** [1]

Abstract

Mobile health (mHealth) applications can provide users with essential and timely feedback. From physical activity suggestions, to stress-reduction techniques, mHealth can provide a wide spectrum of effective treatments. Personalizing these interventions might vastly improve their effectiveness, as individuals vary widely in their response to treatment. An optimal mHealth policy must address the question of *when* to intervene, even as this question is likely to differ between individuals. The high amount of noise due to the in situ delivery of mHealth interventions can cripple the learning rate when a policy only has access to a single user's data. When there is limited time to engage users, a slow learning rate can pose problems, potentially raising the risk that users leave a study. To speed up learning an optimal policy for each user, we propose learning personalized policies via intelligent use of other users' data. The proposed learning algorithm allows us to pool information from other users in a principled, adaptive manner. The algorithm combines Thompson sampling with a Bayesian random effects model for the reward function. We use the data collected from a real-world mobile health study to build a generative model and evaluate the proposed algorithm in comparison with two natural alternatives: learning the treatment policy separately per person and learning a single treatment policy for all people. This work is motivated by our preparations for a real-world followup study in which the proposed algorithm will be used on a subset of the participants.

[*]Equal contribution [1]Harvard Unversity [2]University of Michigan [3]Stanford University. Correspondence to: Sabina Tomkins <sabina_tomkins@fas.harvard.edu>.

Reinforcement Learning for Real Life (RL4RealLife) Workshop in the 36^{th} International Conference on Machine Learning, Long Beach, California, USA, 2019.

1. Introduction

Mobile health (mHealth) interventions deliver treatments to users to support healthy behaviors. For example, to help users increase their physical activity, an mHealth application might send a suggestion to walk when a user is motivated and able to pursue the suggestion. The promise of mHealth interventions hinges on their ability to provide support at times when users need the support and are receptive to it (Nahum-Shani et al., 2017). Thus, in mHealth our goal is to learn an optimal policy of when and how to intervene for a given user and context. A significant challenge to learning an optimal policy is that there are often only a few opportunities per day to provide treatment. Furthermore, there are inherent quality issues with wearable sensors which provide noisy estimates of true step counts (Kaewkannate & Kim, 2016) and with data from mobile phones (e.g. location might note be accurate as users do not always carry their phones). In mHealth settings, a learning algorithm should learn quickly in spite of the noisy data and small number of treatment times. Learning quickly is critical as a poor policy can decrease user engagement. To speed learning, we propose an approach that intelligently pools data from all users so as to more quickly learn an optimal policy for each user. Our approach is adaptive in that we update the relative contribution of individual to population-level data.

The algorithm will be used in a clinical trial for individuals with early stage hypertension. To provide data to design this clinical trial we conducted a physical activity study with sedentary individuals[1], HEARTSTEPS. In HEARTSTEPS each user may receive contextually tailored activity suggestion as a smartphone notification at any of 5 times per day. Contextual data was collected from each user's fitness tracker and smartphone. We evaluate our approach with a simulation environment, which we construct with data collected from the above physical activity study. By mirroring aspects of this study we aim to evaluate the algorithm in a realistic setting in which each user may experience the treatments a few times per day and in which the data is noisy. As these settings extend beyond mHealth and there is a dearth

[1]We are withholding the name of the study to respect the double-blind nature of this submission. It will be released in the final version of this manuscript.

of acceptable methods to contend with their inherent challenges, we propose our approach as a general framework for principled pooling in reinforcement learning algorithms.

2. Approach

Our approach is motivated to meet the needs of mHealth studies. In these settings, users join the study in a staggered fashion, e.g., they enter the study at different times. During the study the developed algorithm will learn a policy for each user based on the user's prior data as well as data available from other current users and past users. Recall that a policy takes the user's current context as input and outputs a treatment such as a physical activity message.

We first introduce the notation used in Section 2.1. In Section 2.2, we formalize our objective of learning individual policies as contextual bandit problems. Then in Section 2.3, we describe how we extend this formulation to adaptively pool the user's data with data available from other current users and past users.

2.1. Problem setting

Let i be the user index. For each user, we use $k \in \{1, 2, \dots\}$ to index the decision times, i.e., times at which an intervention could be provided. Denote by $S_{i,k}$ the contextual variables at the k-th decision time of user i, such as weather, location, time of the day and activity level. Let $A_{i,k}$ be the selected treatment. For simplicity, we consider the binary action space $\mathcal{A} = \{0, 1\}$. Recall that the users enter the study in staggered fashion. We denote by $t_{i,k}$ the calendar time of user i's k-th decision time.

Our objective is to learn individual treatment policies for N individuals; we treat this as N contextual bandit problems. We note that maintaining N separate problems is important in settings such as ours where the true context is only sparsely observed and there is significant unobserved heterogeneity among different users. In the following sections, we describe our Thompson Sampling-based approach for learning the treatment policy for any specific user in the study.

2.2. Separate bandit problem per user

In this section, we consider learning the treatment policy separately per person. At each decision time k, we would like to select an action $A_{i,k} \in \{0, 1\}$ based on the context $S_{i,k}$. To determine how to compute $\pi_{i,k}$, we first model the reward $R_{i,k}$ received from any decision using a Bayesian linear regression model for each user i:

$$R_{i,k} = \phi(S_{i,k}, A_{i,k})^\top \theta_i + \epsilon_{i,k} \tag{1}$$

where ϕ[2] is a mapping such that $\phi(S_t, A_t) \in \mathbb{R}^p$ is the feature of current context and action used to predict the reward, θ_i is a parameter vector which we will learn, and $\epsilon_{i,k} \sim N(0, \sigma_\epsilon^2)$ is the error term. We also specify independent priors across users, $\theta_i \sim N(\mu_\theta, \Sigma_\theta)$ for each user i.

Now at any decision time k, given the history of data so far $\mathcal{H}_{i,k} = \{(S_o, A_o, R_o) : o \leq k\}$ for the user, we take a Thompson Sampling approach to sample the next action. That is, we compute the posterior distribution for θ_i and for context $S_{i,k} = s$, select action $A_{i,k} = 1$ with probability $\pi_{i,k}$

$$\pi_{i,k} = \Pr\{(\phi(s, 1) - \phi(s, 0))^T \tilde{\theta}_i > 0\} \tag{2}$$

where $\tilde{\theta}_i$ follows the posterior distribution given the current data $\mathcal{H}_{i,k}$.

2.3. Intelligent pooling across bandit problems

In many mobile health applications, the combination of noisy data as well as few decision points per day means that learning the treatment policy separately per user can suffer from a slow policy improvement. Our key insight in this work is that we can leverage data collected from other users to improve our ability to learn the optimal treatment policy for each user. To provide some intuition, if we assume all users are identical, e.g., they share the same expected reward function, we could build a single Bayesian regression model:

$$R_{i,k} = \phi(S_{i,k}, A_{i,k})^T \theta + \epsilon_{i,k}.$$

Note that θ does not vary by i. We could then use the posterior distribution of the parameter θ to form a common treatment policy for all users. However, such an approach ("complete pooling") may suffer from high bias when there is significant heterogeneity among users. Instead, our proposed method will pool information across users in an adaptive way, i.e., when there is strong (or weak) heterogeneity observed in the current collected data, the method will pool less (or more) from others while learning the treatment policy.

2.3.1. Bayesian random effects model

We now describe how this pooling model is incorporated into our contextual bandit framework before outlining how we learn the treatment policy for each user. Consider the Bayesian linear regression model (1). Instead of considering the θ_is as separate parameters to be estimated, we impose a structure on θ_i:

$$\theta_i = \theta_{pop} + u_i \tag{3}$$

[2] Defined in Section 3.2

θ_{pop} is a population-level parameter and u_i represents the person-specific deviation from θ_{pop} for user i, as in a standard random effects model (Raudenbush & Bryk, 2002; Laird et al., 1982) We use the following prior for this model: (1) θ_{pop} has prior mean μ_θ and variance Σ_θ, (2) u_i has mean $\mathbf{0}$ and variance Σ_u, (3) $u_i \perp\!\!\!\perp u_j$ for $i \neq j$ and $\theta_{pop} \perp\!\!\!\perp \{u_i\}$. $\mu_\theta, \Sigma_\theta$ as well as the variance of the person-specific effect Σ_u and the residual variance σ_ϵ^2 are hyper-parameters. These may be determined either using domain knowledge or learned from prior data.

We denote by $\mathcal{T}$ the set of times that the posterior distribution is updated. Specifically, let $T \in \mathcal{T}$ be an updating time and $\mathcal{U}$ be the set of users that are currently in the study or have finished the study. The data available at time T is $\mathcal{D}_T = \{(S_{i,k}, A_{i,k}, R_{i,k}, i, k) : i \in \mathcal{U}, t_{i,k} \leq T\}$. The posterior distribution of each $\theta_{i,k}$ is Gaussian with mean and variance determined by a kernel function induced by the mixed effects model (Eqns. 1, 3): for any two tuples $x_l = (S^{(l)}, A^{(l)}, R^{(l)}, i_l, k_l), l = 1, 2,$

$$k_\lambda(x_1, x_2) = \phi_1^\top \Sigma_\theta \phi_2 + 1_{i_1=i_2} \phi_1^\top \Sigma_u \phi_2$$

where $\phi_l = \phi(S^{(l)}, A^{(l)})$. Suppose the number of tuples in the training data $\mathcal{D}_T$ is n_T. The kernel matrix K_λ is of size $n_T \times n_T$ and each element is the kernel value between two tuples in $\mathcal{D}_T$. The posterior mean and variance of θ_i given $\mathcal{D}_T$ can be calculated by

$$\mu_i = \mu_\theta + M_i^\top (K_\lambda + \sigma_\epsilon^2 I_T)^{-1} \tilde{R}_{n_T}$$
$$\Sigma_i = \Sigma_\theta + \Sigma_u - M_i^\top (K_\lambda + \sigma_\epsilon^2 I_{n_T})^{-1} M_i$$

where $\tilde{R}_{n_T}$ is the vector of the rewards centered by the prior means, i.e., each element corresponds to a tuple (S, A, R, j, h) in $\mathcal{D}_T$ and is given by $R - \phi(S, A)^\top \mu_\theta$, and M_i is a matrix of size n_T by p, with each row corresponding to a tuple (S, A, R, j, h) in $\mathcal{D}_T$ and given by $\phi(S, A)^\top (\Sigma_\theta + 1_{j=i} \Sigma_u)$.

2.3.2. Action selection

To select the action for user i at the k-th decision time, we use the posterior distribution of $\theta_{i,k}$ formed at the most recent update time, T. That is for context, $S_{i,k} = s$, select the action for user i at this user's k-th decision time by

$$\pi_{i,k} = \Pr\{(\phi(s, 1) - \phi(s, 0))^T \tilde{\theta}_{i,k} > 0\} \qquad (4)$$

where $\tilde{\theta}_{i,k}$ follows the posterior distribution.

2.3.3. Updating hyper-parameters at update times

Thus far we have described how to learn individual treatment policies while pooling across users and contexts which are similar as determined by the hyper-parameters. However we can update or re-adjust the degree of pooling from different users by re-estimating hyper-parameters as we collect more data from users. While the prior mean and prior variance of the population parameters θ_{pop} can be set according to previous study or domain knowledge, it is difficult to pretune the variance components in the random effect. Here we use an Empirical Bayes (Carlin & Louis, 2010) approach to update Σ_u as well as σ_ϵ^2 by choosing the values that maximize the marginal log-likelihood of the observed reward, marginalized over the population parameters θ_{pop} and the random effects. In other words, at every update time, T, we set the hyper-parameters as $\hat{\lambda} = \text{argmax}\, l(\lambda|\mathcal{D}_T)$ where $\lambda = (\Sigma_u, \sigma_\epsilon^2)$ and the marginal likelihood $l(\lambda|\mathcal{D}_T)$ is given by

$$l(\lambda|\mathcal{D}_T) = -\frac{1}{2}[\tilde{R}_{n_T}^\top (K_\lambda + \sigma_\epsilon^2 I_{n_T})^{-1} \tilde{R}_{n_T} + \log\det(K_\lambda + \sigma_\epsilon^2 I_{n_T}) + n_T \log(2\pi)]$$

This full algorithm, including the updates of the pooling hyper-parameters, is summarized in Algorithm 1.

Algorithm 1 Thompson Sampling with Intelligent Pooling

for $t \in [0, T]$ **do**
  Receive user index i and the decision time index k
  Collect the states variable S and availability indicator I
  if $I = 1$ **then**
    Obtain posterior distribution $post(i|\mathcal{D}, \lambda)$
    Calculate the randomization probability π.
    Sample the action $A \sim \text{Bern}(\pi)$
    Collect the reward R
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{S, A, R, i, k\}$
  end if
  if $t \in \mathcal{T}$ **then**
    Update the hyper-parameters: $\lambda \leftarrow \text{argmax}\, l(\lambda|\mathcal{D})$
    Update the posterior distribution: $post(\cdot) = post(\cdot|\mathcal{D}, \lambda)$
  end if
end for

3. Experiments

This algorithm is designed for an ongoing multi-stage trial of an mHealth physical activity study. Thus to evaluate our approach under as realistic as possible conditions, we construct a simulation environment from a prior stage of the study which we refer to throughout as HeartSteps. This simulation allows us to not only anticipate and solve many of the difficulties of deploying an adaptive learning algorithm in real-time, but also to explore various settings under which users share underlying characteristics. In Section 3.1, we first describe our simulation environment. Then we describe how we generate users' step counts within this simulation environment. In Section 3.2 we provide implementation details, and finally in Section 3.3 we present empirical results for our algorithm in the simulation environment.

3.1. Simulation environment

We construct a simulation environment using data from a prior mHealth study (HEARTSTEPS(Klasnja et al., 2018).) in which participants used a mobile phone application along with a wristband fitness tracker. We describe this simulation environment with two data types: a TRIAL and a USER. Each data type possesses both static characteristics and dynamic variables.

The context $S_{i,k}$ of user i at time k can be expressed as functions of both static characteristics and dynamic context features. Step counts are generated for each USER every thirty minutes and the reward $R_{i,k}$ is the step count in the thirty minutes immediately following an intervention. We first describe the static characteristics of a TRIAL and a USER in Section 3.1.1 and then describe the dynamic context features of each in Section 3.1.2.

3.1.1. STATIC CHARACTERISTICS

TRIAL characteristics A TRIAL is described by both static characteristics, which are set at initialization and consequently remain the same, and dynamic context features which change over time. The static characteristics are: the number of users who will participate in the study, the decision times at which an intervention might be sent, and the recruitment rate at which users will join the study.

USER characteristics A USER is also described by static characteristics and dynamic context features. The static characteristics are: the day at which the USER enters the TRIAL, the total time the USER will remain in the TRIAL, and the USER's general physical activity levels. A USER's base activity level is determined by assignment to one of two groups: low-activity participants or high-activity participants. These two groups were discovered from HEARTSTEPS by performing non-parametric clustering where two groups were found to best fit the data. When a USER joins the study they are placed into either group one or two with equal probability. Group membership is not known to the RL algorithm.

3.1.2. RUNNING THE SIMULATION

Both aspects of the TRIAL's context and of USERS' context are updated dynamically. Every thirty minutes the current date and time are updated. At each time we form features of the TRIAL which are common across all active users at this time. Additionally, we form USER-specific features. These dynamic features are outlined in Table 1.

Each of the features in Table 1 stems from HEARTSTEPS. We used domain science to inform the feature design as much as possible. Here, the location feature was informed by domain experts. In the other cases we constructed the features in order to best explain different levels of physical activity, according to HEARTSTEPS. To choose how to partition the day into meaningful segments we evaluate different partitions on their ability to form separable clusters of activity levels. For example, we anticipate that given the right partition each time of day segment would have different average observed activity levels, as they would each explain activity at different contextually meaningful times, e.g. activity at night should be different than at midday. Similarly, we considered different groupings of days of the week, and different numbers of partitions for temperature, and for preceding activity levels. For both temperature and preceding activity levels we found two groups to best explain the observed step counts in HEARTSTEPS, and to discretize these continuous values we used the median temperature and step-counts from HEARTSTEPS respectively. All of these feature representations were informed from HEARTSTEPS.

Common across the **TRIAL**	
Name	**Value**
Time of day	Morning(0) - 9:00 and 15:00 Afternoon(1) - 15:00 and 21:00 Night(2) -21:00 and 9:00
Day of the week	Weekday(0) or Weekend(1)
Temperature	Cold(0) or Hot(1)
Specific to each **USER**	
Name	**Value**
Preceding activity level	Low(0) or High(1)
Location	Other(0) or Home/work(1)
Available	No(0) or Yes(1)

Table 1. Dynamic features describing both TRIAL and USER states. The value used in encoding each variable is shown in parentheses. For example cold(0) indicates that cold is coded as a 0 wherever this feature is used.

Both temperature and location are updated five times a day, this choice arises from HEARTSTEPS , where we have readings for temperature and location 5 times a day. These variables are updated roughly every two hours from 9:00 to 19:00. Each new temperature is generated as a function of the current month and the current temperature. Each new location is a function of a USER's group-id, the time of day, the day of the week, and their current location. To capture the fact that USERS are not always available (usually due to operating a motor vehicle) to receive treatment we introduce the context feature $Available \sim Bernoulli(.8)$. A user's availability to receive treatment is updated at each decision time.

A simulation runs for the course of a TRIAL until all recruited USERS have finished the study. Every thirty-minutes from the beginning to the end of the study dynamic TRIAL variables are updated as well as the dynamic variables for all active USERS. A new step-count is generated for each USER active in the study, every thirty-minutes according to one of the following scenarios:

1. USER is at a decision time
   (a) USER is available

(b) USER is not available

2. USER is not at a decision time

Scenarios 1b and 2 are equivalent with respect to how step-counts are generated; a USER's step count either depends on whether or not they received an intervention (when they are at a decision time and available) or it does not (because they were either not at a decision time or not available).

To generate step counts we obtain sufficient statistics from HEARTSTEPS. Here i denotes the ith user and k denotes the time of day. Consider a function $h(S_{i,k})$ which selects all aspects of context $S_{i,k}$ which are relevant in generating a step count and which forms a vector of discrete context values $h(S_{i,k}) \in \{0,1\}^n$. Here, $h(S_{i,k})$ contains the group-id of user i, the time of day at time k, the day of the week at time k, the temperature at time k, the preceding activity level of user i at time k and the location of user i at time k. For each possible $h(S_{i,k})$ we obtain $\mu_{S_{i,k}}$ and $\sigma_{S_{i,k}}$, where $\mu_{h(S_{i,k})}$ and $\sigma_{h(S_{i,k})}$ are the empirical mean and standard deviation of all step counts observed when $h(S_{i,k})$ is encountered in HEARTSTEPS. We introduce the function $f(S_{i,k})$ which selects only those variables which are included in the reward model for a particular algorithm. Let β be a vector of context coefficients which weigh the relative contributions of the entries of $f(S_{i,k})$ to the reward. We find the magnitude of the entries of β from HEARTSTEPS. We then generate step counts ($R_{i,k}$) at decision times when users are available according to Equation 5 (Scenario 1a). Under either Scenarios 1b or 2, we generate step counts from Equation 6.

$$R_{i,k} = \mathcal{N}(\mu_{h(S_{i,k})}, \sigma_{h(S_{i,k})}) + A_{i,k}(f(S_{i,k})^T\beta + Z_i) \quad (5)$$

$$R_{i,k} = \mathcal{N}(\mu_{h(S_{i,k})}, \sigma_{h(S_{i,k})}), \quad (6)$$

The variable, Z_i is the person-specific effect for the ith user.

If a USER is at a decision time and is available they will receive a treatment according to whichever RL policy is being run through the simulation. A policy is an independent input to the simulation.

3.2. Implementation details

TRIAL implementation details Each study has 32 USERS. For simplicity, in these experiments the recruitment rate is set so that all users join the study on the first day. As each USER remains in the study for 12 weeks, the entire length of the study is 12 weeks. The decision times are set roughly two hours apart from 9:00 to 19:00.

We consider three scenarios (shown in Table 2) to generate Z_i, the person-specific effect, the performance of each algorithm under each scenario will be analyzed in Section 3.3. We design z_1 and z_2 so that for all users in group 1, it is optimal to send a message 75% of the time while for all users in group 2 it is optimal to send a message 25% of the time. Recall that the bandit algorithm will not have access to group membership. We set σ as the standard deviation of the observed treatment effects $[f(S_{i,k})^\top\beta \ : \ S_{i,k} \in \text{HEARTSTEPS}]$.

Homogeneous	Bi-modal	Smooth
$Z^i = 0$	$Z_i = \begin{cases} z_1, & \text{if } i \in \text{group one} \\ z_2, & \text{if } i \in \text{group two} \end{cases}$	$Z_i \sim \mathcal{N}(0, \sigma^2)$

Table 2. Settings for Z in three cases of homogeneous, bimodal and smoothly varying populations.

Policy learning implementation details In Section 2 we introduced the feature vector ϕ, recall that ϕ is a mapping $\phi(S_{i,k}, A_{i,k}) \in \mathbb{R}^p$. To define ϕ we first introduce $g(S_{i,k})$, a vector which selects context information relevant to generating baseline step counts. This is different from $f(S_{i,k})$ which captures context variables relevant to how responsive the user is to treatment. For example, one's general activity level might depend on their overall physical activity earlier in the day while their responsivity might depend on their current location. The vector $g(S_{i,k})$ is a subset of $h(S_{i,k})$, containing: time of day, day of the week, preceding activity level, and location. The entries to $f(S_{i,k})$ are: the preceding activity level and location for user i at time k. Let $\pi_{i,k}$ be a probability of treatment. Then

$$\phi(S_{i,k}, A_{i,k}) = [g(S_{i,k}, A_{i,k}), \pi_{i,k} f(S_{i,k}), (A_{i,k}\pi_{i,k}) f(S_{i,k})].$$

This choice to include the term $(A_{i,k} - \pi_{i,k})f(S_{i,k})$ is motivated by (Liao et al., 2016; Boruvka et al., 2018; Greenewald et al., 2017), who demonstrated that action-centering can protect against mis-specification in the baseline effect (e.g., the expected reward under the action 0).

For simplicity we put the user-specific effect only on the intercept terms in both the baseline and treatment effect models. Finally, we constrain the randomization probability to be within [0.1, 0.8] to ensure continual learning. The update time for the hyper-parameters is set to be every 7 days. All approaches are implemented in Python [3]. We implement the GP regression with the software package GPytorch (Gardner et al., 2018).

3.3. Empirical evaluation

In this section, we present an empirical analysis of our algorithm compared to two competing methods. We refer to the three approaches as: COMPLETE, PERSON-SPECIFIC and INTELLIGENTPOOLING. In the first approach, COMPLETE, we treat all individuals the same and learn one set of parameters across the entire population, pooling the entire dataset in an unstructured way. This allows us to use all available data, but does not allow the learning of individual-level parameters (the u_i's). Alternatively, we compare to

[3] https://github.com/StatisticalReinforcementLearningLab/intelligent_pooling

PERSON-SPECIFIC, the person-specific approach outlined in Section 2.2.

First, we show the ability of each algorithm to select the correct action at each decision time. For each decision time, we compute the fraction of actions which were optimal across all participants, denoted as ρ^*. In Fig. 1 we show ρ^* averaged across 50 simulations for each decision time. Additionally, we present the results of these algorithms in terms of post-treatment step counts, that is we consider the steps taken in the thirty-minutes following an intervention suggested by a given learning algorithm. We consider the effectiveness of each algorithm with respect to the various underlying models of treatment effects. Here, we consider the three setting shown in Table 2. Throughout this section we evaluate on a population of 32 users.

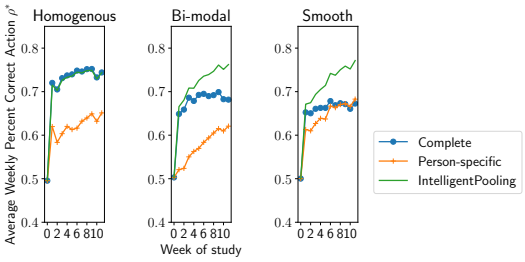

Figure 1. ρ^* averaged over 50 simulations each week. Hyper-parameters updated weekly.

We additionally show the ability of each algorithm to select the correct action at each decision time. For each decision time, we compute the fraction of actions which were optimal across all participants, denoted as ρ^*. In Fig. 1 we show the ρ^* averaged across 50 simulations for each week of the simulated study.

In the Bi-modal setting there are two groups, where all users in group one have a negative response to treatment on average, while the users in the other group have a positive response to treatment. An optimal policy would learn to not send interventions to users in the first group, and to send them to users in the second. To evaluate the extent to which each algorithm can learn this distinction we show the percentage of time each group received a message.

	Homogeneous Z^h	Bi-modal Z^b	Smooth Z^s
COMPLETE	4.85 (.0018)	4.65 (.0018)	4.86 (.0019)
PERSON-SPECIFIC	4.80 (.0018)	4.60 (.0019)	4.85 (.0019)
INTELLIGENT-POOLING	4.85 (.0018)	4.66 (.0018)	4.89 (.0019)

Table 3. Average post-treatment step count under each algorithm. Thirty-minute step count shown in log scale. Results averaged over 50 simulations. Standard errors of the mean shown in parentheses.

	Group one optimal policy = send	Group one optimal policy = don't send
COMPLETE	0.52	0.49
PERSON-SPECIFIC	0.68	0.49
INTELLIGENT-POOLING	0.56	0.42

Table 4. Average fraction of times message was sent (action=1), over 50 simulations (bi-modal generative model Z^b).

4. Discussion

In our empirical evaluation we compared three approaches for learning action-selection policies in an mHealth setting. For each approach, we consider three population settings with differing generative models for person-specific treatment effects. We expect the performance of each approach to be related to the configurations of the underlying generative model.

We first analyze the homogenous setting where there are no differences between users' underlying generative models ($Z_i = 0$ for all users i). Both COMPLETE and INTELLIGENTPOOLING achieve the highest overall average post-treatment step count. This is mirrored in Fig. 1. PERSON-SPECIFIC suffers from a low amount of data and the its performance remains below that of COMPLETE and INTELLIGENTPOOLING as shown in Fig. 1.

In the bi-modal setting all participants in group one are given a fixed $Z_i = 0$ where all participants in group two are given $Z_i = -0.5$. This enforces that the optimal policy for anyone in group one would be to send a message most of the time, while for all those in group two the optimal policy would be to not send a message most of the time. In Table 3 we do see that INTELLIGENTPOOLING achieves higher average post-treatment step count than COMPLETE. To understand the performance of PERSON-SPECIFIC inspect Fig. 1. In Fig. 1 we see that the performance of both PERSON-SPECIFIC and INTELLIGENTPOOLING increases with time, while the performance of COMPLETE plateaus early. While INTELLIGENTPOOLING surpasses COMPLETE by the first week, PERSON-SPECIFIC improves much more slowly, it is not until the eighth week that PERSON-SPECIFIC meets the performance of COMPLETE.

In the smooth setting all participants receive a unique treatment effect $Z_i \sim \mathcal{N}(0, \sigma) \forall i$. Here, we expect INTELLIGENTPOOLING to achieve the highest performance as its underlying assumptions best match the generative model. We do see that INTELLIGENTPOOLING both achieves the highest average post-treatment step count and the highest percent of correct actions chosen.

Additionally, we are interested to see if the approaches are able to learn different policies for each group. To analyze this, we consider the bi-modal setting where there are two

differing optimal policies, one for each group of participants. In Table 4 we show the average number of times an intervention was sent for each group of participants across 50 simulations. Here we see that both PERSON-SPECIFIC and INTELLIGENTPOOLING were able to better differentiate the two groups.

In mHealth settings it is difficult to get the high quality data required to learn a good policy in a complicated state space. Here, we offer first steps in addressing the challenges of this domain. We find that the choice of algorithm depends on the underlying characteristics of a given population. When participants are completely homogenous there seems to be no disadvantage to pooling completely and learning one optimal policy for the population. However, as the population varies, and even if there are at least two distinguishable groups (which we did find to be the case in HEARTSTEPS) there are advantages to pooling intelligently and retaining some personalization in the policies for each user. Depending on the characteristics of a population there are many choices which can guide the education of an optimal policy.

5. Related Work

In mHealth several algorithms have been proposed for learning treatment policies. These have typically followed two main paradigms. The first is learning a treatment policy for each user separately, such as (Rabbi et al., 2015), (Jaimes et al., 2016), and (Forman et al., 2018). This approach makes sense when users are highly heterogeneous, that is, their optimal policies differ greatly one from another. However, this can present challenges for learning the policy when data is scarce and/or noisy, as in our motivating example of encouraging activity in an mHealth study where only a few decision time-points occur each day. The second paradigm is learning one treatment policy for all users: a bandit algorithm is used in (Bouneffouf et al., 2012; Paredes et al., 2014; Yom-Tov et al., 2017), and (Clarke et al., 2017) and (Zhou et al., 2018) consider a full reinforcement learning algorithm. This second approach can potentially learn quickly but may result in poor outcomes if the optimal policies differ much between users. In this work, we learn a separate treatment policy for each user. However our proposed algorithm adaptively pools information across users and is thus less reliant on each individual's noisy data.

The proposed algorithm uses a mixed (random) effects Gaussian process (GP) model as part of a Thompson-Sampling algorithm. Gaussian process models have been used in (Chowdhury & Gopalan, 2017; Brochu et al., 2010; Srinivas et al., 2009; Desautels et al., 2014; Wang et al., 2016; Djolonga et al., 2013; Bogunovic et al., 2016) for multi-armed bandits, and in (Zhou, 2015; Li et al., 2010; Krause & Ong, 2011) for contextual bandits. Our use of a mixed-effects GP builds off of work such as (Shi et al., 2012; Luo et al., 2018) in the prediction setting; however we consider a mixed-effects model in the context of reinforcement learning. Our algorithm adaptively updates the degree to which other users' data is used to learn each user's policy, with the inclusion of mixed effects and by updating of the variance hyper-parameters.

Several existing works in the bandit literature use pooling in other aspects of the model: (Deshmukh et al., 2017) pools data from different arms of a single bandit, and (Li & Kar, 2015) uses context-sensitive clustering to produce aggregate reward estimates for the UCB bandit algorithm. More relevant to our work are multi-task Gaussian processes, e.g. (Lawrence & Platt, 2004; Yu et al., 2005; Bonilla et al., 2008), though these have been investigated in the prediction as opposed to reinforcement learning setting. (Bonilla et al., 2008) modulates the Gaussian process covariance function over inputs with an additional inter-task similarity matrix. (Wang & Khardon, 2012) connect mixed-effects models to GP multitask learning (Wang & Khardon, 2012), however not in a reinforcement learning context. We also use a Gaussian process-based approach for pooling in our method; however in contrast to these prior works, we specifically personalize reinforcement learning bandits. Similar to multi-task learning, with meta-learning one exploits shared structure across tasks to improve performance on new tasks. Our approach thus shares similarities with meta-learning for reinforcement learning (Nagabandi et al., 2018; Finn et al., 2019; 2018; Zintgraf et al., 2019; Gupta et al., 2018; Sæmundsson et al., 2018). While meta-learning might require a large collection of source tasks, we demonstrate the efficacy of our approach on data on the same small scale as that found in mHealth studies.

6. Conclusion

We have introduced a general methodology for intelligent pooling in Thompson sampling algorithms. While here we evaluate the algorithm using a small number of random effects, the method is naturally generalizable to the inclusion of additional random effects. One hindrance in mobile health studies is an increased lack of engagement as the study continues and participants become over-burdened. A natural extension to our current approach would be a full reinforcement learning algorithm that incorporates the delayed effects of treatment.

References

Bogunovic, I., Scarlett, J., and Cevher, V. Time-varying gaussian process bandit optimization. In *Artificial Intelligence and Statistics*, pp. 314–323, 2016.

Bonilla, E. V., Chai, K. M., and Williams, C. Multi-task gaussian process prediction. In *Advances in neural infor-*

mation processing systems, pp. 153–160, 2008.

Boruvka, A., Almirall, D., Witkiewitz, K., and Murphy, S. A. Assessing time-varying causal effect moderation in mobile health. *Journal of the American Statistical Association*, 113(523):1112–1121, 2018.

Bouneffouf, D., Bouzeghoub, A., and Gançarski, A. L. Hybrid-ε-greedy for mobile context-aware recommender system. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 468–479. Springer, 2012.

Brochu, E., Hoffman, M. W., and de Freitas, N. Portfolio allocation for bayesian optimization. *arXiv preprint arXiv:1009.5419*, 2010.

Carlin, B. P. and Louis, T. A. *Bayes and empirical Bayes methods for data analysis*. Chapman and Hall/CRC, 2010.

Chowdhury, S. R. and Gopalan, A. On kernelized multi-armed bandits. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 844–853. JMLR. org, 2017.

Clarke, S., Jaimes, L. G., and Labrador, M. A. mstress: A mobile recommender system for just-in-time interventions for stress. In *2017 14th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, pp. 1–5. IEEE, 2017.

Desautels, T., Krause, A., and Burdick, J. W. Parallelizing exploration-exploitation tradeoffs in gaussian process bandit optimization. *The Journal of Machine Learning Research*, 15(1):3873–3923, 2014.

Deshmukh, A. A., Dogan, U., and Scott, C. Multi-task learning for contextual bandits. In *Advances in Neural Information Processing Systems*, pp. 4848–4856, 2017.

Djolonga, J., Krause, A., and Cevher, V. High-dimensional gaussian process bandits. In *Advances in Neural Information Processing Systems*, pp. 1025–1033, 2013.

Finn, C., Xu, K., and Levine, S. Probabilistic model-agnostic meta-learning. In *Advances in Neural Information Processing Systems*, pp. 9516–9527, 2018.

Finn, C., Rajeswaran, A., Kakade, S., and Levine, S. Online meta-learning. *arXiv preprint arXiv:1902.08438*, 2019.

Forman, E. M., Kerrigan, S. G., Butryn, M. L., Juarascio, A. S., Manasse, S. M., Ontañón, S., Dallal, D. H., Crochiere, R. J., and Moskow, D. Can the artificial intelligence technique of reinforcement learning use continuously-monitored digital data to optimize treatment for weight loss? *Journal of behavioral medicine*, pp. 1–15, 2018.

Gardner, J., Pleiss, G., Weinberger, K. Q., Bindel, D., and Wilson, A. G. Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. In *Advances in Neural Information Processing Systems*, pp. 7576–7586, 2018.

Greenewald, K., Tewari, A., Murphy, S., and Klasnja, P. Action centered contextual bandits. In *Advances in neural information processing systems*, pp. 5977–5985, 2017.

Gupta, A., Mendonca, R., Liu, Y., Abbeel, P., and Levine, S. Meta-reinforcement learning of structured exploration strategies. In *Advances in Neural Information Processing Systems*, pp. 5302–5311, 2018.

Jaimes, L. G., Llofriu, M., and Raij, A. Preventer, a selection mechanism for just-in-time preventive interventions. *IEEE Transactions on Affective Computing*, 7(3): 243–257, 2016.

Kaewkannate, K. and Kim, S. A comparison of wearable fitness devices. *BMC public health*, 16(1):433, 2016.

Klasnja, P., Smith, S., Seewald, N. J., Lee, A., Hall, K., Luers, B., Hekler, E. B., and Murphy, S. A. Efficacy of contextually tailored suggestions for physical activity: A micro-randomized optimization trial of heartsteps. *Annals of Behavioral Medicine*, 2018.

Krause, A. and Ong, C. S. Contextual gaussian process bandit optimization. In *Advances in Neural Information Processing Systems*, pp. 2447–2455, 2011.

Laird, N. M., Ware, J. H., et al. Random-effects models for longitudinal data. *Biometrics*, 38(4):963–974, 1982.

Lawrence, N. D. and Platt, J. C. Learning to learn with the informative vector machine. In *Proceedings of the twenty-first international conference on Machine learning*, pp. 65. ACM, 2004.

Li, L., Chu, W., Langford, J., and Schapire, R. E. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pp. 661–670. ACM, 2010.

Li, S. and Kar, P. Context-aware bandits. *arXiv preprint arXiv:1510.03164*, 2015.

Liao, P., Klasnja, P., Tewari, A., and Murphy, S. A. Sample size calculations for micro-randomized trials in mhealth. *Statistics in medicine*, 35(12):1944–1971, 2016.

Luo, L., Yao, Y., Gao, F., and Zhao, C. Mixed-effects gaussian process modeling approach with application in injection molding processes. *Journal of Process Control*, 62:37–43, 2018.

Nagabandi, A., Finn, C., and Levine, S. Deep online learning via meta-learning: Continual adaptation for model-based rl. *arXiv preprint arXiv:1812.07671*, 2018.

Nahum-Shani, I., Smith, S. N., Spring, B. J., Collins, L. M., Witkiewitz, K., Tewari, A., and Murphy, S. A. Just-in-time adaptive interventions (jitais) in mobile health: key components and design principles for ongoing health behavior support. *Annals of Behavioral Medicine*, 52(6): 446–462, 2017.

Paredes, P., Gilad-Bachrach, R., Czerwinski, M., Roseway, A., Rowan, K., and Hernandez, J. Poptherapy: Coping with stress through pop-culture. In *Proceedings of the 8th International Conference on Pervasive Computing Technologies for Healthcare*, pp. 109–117. ICST (Institute for Computer Sciences, Social-Informatics and . . . , 2014.

Rabbi, M., Aung, M. H., Zhang, M., and Choudhury, T. Mybehavior: automatic personalized health feedback from user behaviors and preferences using smartphones. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pp. 707–718. ACM, 2015.

Raudenbush, S. W. and Bryk, A. S. *Hierarchical linear models: Applications and data analysis methods*, volume 1. Sage, 2002.

Sæmundsson, S., Hofmann, K., and Deisenroth, M. P. Meta reinforcement learning with latent variable gaussian processes. *arXiv preprint arXiv:1803.07551*, 2018.

Shi, J., Wang, B., Will, E., and West, R. Mixed-effects gaussian process functional regression models with application to dose–response curve prediction. *Statistics in medicine*, 31(26):3165–3177, 2012.

Srinivas, N., Krause, A., Kakade, S. M., and Seeger, M. Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv preprint arXiv:0912.3995*, 2009.

Wang, Y. and Khardon, R. Nonparametric bayesian mixed-effect model: A sparse gaussian process approach. *arXiv preprint arXiv:1211.6653*, 2012.

Wang, Z., Zhou, B., and Jegelka, S. Optimization as estimation with gaussian processes in bandit settings. In *Artificial Intelligence and Statistics*, pp. 1022–1031, 2016.

Yom-Tov, E., Feraru, G., Kozdoba, M., Mannor, S., Tennenholtz, M., and Hochberg, I. Encouraging physical activity in patients with diabetes: intervention using a reinforcement learning system. *Journal of medical Internet research*, 19(10), 2017.

Yu, K., Tresp, V., and Schwaighofer, A. Learning gaussian processes from multiple tasks. In *Proceedings of the 22nd International Conference on Machine Learning (ICML-05)*, pp. 1012–1019, 2005.

Zhou, L. A survey on contextual multi-armed bandits. *arXiv preprint arXiv:1508.03326*, 2015.

Zhou, M., Mintz, Y., Fukuoka, Y., Goldberg, K., Flowers, E., Kaminsky, P., Castillejo, A., and Aswani, A. Personalizing mobile fitness apps using reinforcement learning. In *IUI Workshops*, 2018.

Zintgraf, L. M., Shiarlis, K., Kurin, V., Hofmann, K., and Whiteson, S. CAML: Fast context adaptation via meta-learning, 2019. URL https://openreview.net/forum?id=BylBfnRqFm.