Machine Learning (2021) 110:2685–2727
https://doi.org/10.1007/s10994-021-05995-8

IntelligentPooling: practical Thompson sampling for mHealth

Sabina Tomkins[1] · Peng Liao[2] · Predrag Klasnja[3] · Susan Murphy[2]

Received: 16 May 2020 / Revised: 10 December 2020 / Accepted: 11 May 2021 /
Published online: 21 June 2021

Abstract

In mobile health (mHealth) smart devices deliver behavioral treatments repeatedly over time to a user with the goal of helping the user adopt and maintain healthy behaviors. Reinforcement learning appears ideal for learning how to optimally make these sequential treatment decisions. However, significant challenges must be overcome before reinforcement learning can be effectively deployed in a mobile healthcare setting. In this work we are concerned with the following challenges: (1) individuals who are in the same context can exhibit differential response to treatments (2) only a limited amount of data is available for learning on any one individual, and (3) non-stationary responses to treatment. To address these challenges we generalize Thompson-Sampling bandit algorithms to develop IntelligentPooling. IntelligentPooling learns personalized treatment policies thus addressing challenge one. To address the second challenge, IntelligentPooling updates each user's degree of personalization while making use of available data on other users to speed up learning. Lastly, IntelligentPooling allows responsivity to vary as a function of a user's time since beginning treatment, thus addressing challenge three.

Keywords Thompson sampling · Mobile health · Clinical trial · Physical activity · Non-stationary environment · Mixed effects · Bayesian reward model

Editors: Yuxi Li, Alborz Geramifard, Lihong Li , Csaba Szepesvari, Tao Wang.

✉ Sabina Tomkins
stomkins@stanford.edu

Peng Liao
pengliao@g.harvard.edu

Predrag Klasnja
klasnja@umich.edu

Susan Murphy
samurphy@fas.harvard.edu

[1] Stanford University, Stanford, United States of America

[2] Harvard University, Cambridge, United States of America

[3] University of Michigan, Ann Arbor, United States of America

1 Introduction

Mobile health (mHealth) applications deliver treatments in users' everyday lives to support healthy behaviors. These mHealth applications offer an opportunity to impact health across a diverse range of domains from substance use (Rabbi et al. 2017), to disease self-management (Hamine et al. 2015) to physical inactivity (Consolvo et al. 2008). For example, to help users increase their physical activity, an mHealth application might send walking suggestions at the times and in the contexts (e.g. current location or recent physical activity) when a user is likely to be able to pursue the suggestions. A goal of mHealth applications is to provide treatments in contexts in which users need support *while* avoiding over-treatment. Over-treatment can lead to user disengagement (Nahum-Shani et al. 2017), for example users might ignore treatments or even delete the application. Consequently, the goal is to be able to learn an optimal policy for when and how to intervene for each user and context without over-treating.

Contextual bandit algorithms appear ideal for this task. Contextual bandit algorithms have been successful in a range of application settings from news recommendations (Li et al. 2010) to education (Qi et al. 2018). However, as we discuss below, many challenges remain to adapt contextual bandit algorithms for mHealth settings. Thompson sampling offers an attractive framework for addressing these challenges. In their seminal work (Agrawal and Goyal 2013), Agrawal and Goyal show that Thompson sampling for contextual bandits, which works well in practice, can also achieve strong theoretical guarantees. In our work, we propose Thompson sampling contextual bandit algorithm which introduces a mixed effects structure for the weights on the feature vector, an algorithm we call IntelligentPooling. We demonstrate empirically that IntelligentPooling has many advantages. We also derive a high-probability regret bound for our approach which achieves similar regret to (Agrawal and Goyal 2013). Unlike (Agrawal and Goyal 2013), our regret bound depends on the variance components introduced by the mixed effects structure which is at the center of our approach.

1.1 Challenges

There are significant challenges to learning optimal policies in mHealth. This work primarily addresses the challenge of learning personalized user policies from limited data. Contextual bandit algorithms can be viewed as algorithms that use the user's context to *adapt* treatment. While this approach can have advantages compared to ignoring the user's context, it fails to address that users can respond differentially to treatments even when they appear to be in the same context. This occurs since sensors on smart devices are unlikely to record all aspects of a user's context that affect their health behaviors. For example, the context may not include social constraints on the user (e.g., care-giving responsibilities), which may influence the user's ability to be active. Thus, algorithms that can learn from the differential responsiveness to treatment are desirable. This motivates the need for an algorithm that not only incorporates contextual information, but that can also learn personalized policies. A natural first approach would be to use the algorithm separately for each user, but the algorithm is likely to learn very slowly if data on a user is sparse and/or noisy. However, typically in mHealth studies multiple users are using the application at any given time. Thus an algorithm that pools data over users intelligently so as to speed up learning of personalized policies is desirable.

An additional challenge is non-stationary responses to treatment (e.g. non-stationary reward function). For example, in the beginning of a study, a user might be excited to receive a treatment, however after a few weeks this excitement can wane. This motivates the need for algorithms that can learn time-varying treatment policies.

1.2 Contributions

We develop IntelligentPooling, a type of Thompson sampling contextual bandit algorithm specifically designed to overcome the above challenges. Our main contributions are:

- IntelligentPooling: *A Thompson sampling contextual bandit algorithm for rapid personalization in limited data settings*. This algorithm employs classical random effects in the reward function (Raudenbush and Bryk 2002; Laird and Ware 1982) and empirical (Bayes Morris 1983; Casella 1985) to adaptively adjust the degree to which policies are personalized to each user. We present an analysis of this adaptivity in Sect. 3.5 showing that IntelligentPooling can learn to personalize to a user as a function of the observed variance in the treatment effect both between and within users.
- A high probability regret bound for IntelligentPooling.
- *An empirical evaluation of* IntelligentPooling *in a simulation environment constructed from mHealth data*. IntelligentPooling not only achieves 26% lower regret than state-of-the-art approaches, it also is better able to adapt to the degree of heterogeneity present in a population than this approach.
- *Feasibility of* IntelligentPooling *from a pilot study in a live clinical trial*. We demonstrate that IntelligentPooling can be executed in a real-time online environment and show preliminary evidence of this method's effectiveness.
- We show how to modify IntelligentPooling to learn in non-stationary environments.

Next, in Sect. 2 we discuss relevant related work. In Sect. 3 we present IntelligentPooling and provide a high-probability regret bound for this algorithm. We then describe how we use historical data to construct a simulation environment and evaluate our approach against state-of-the-art in Sect. 4. Next, in Sect. 5 we introduce the feasibility study and provide preliminary evidence into the benefits of this approach. We then discuss how to extend this work to include time-varying effects in Sect. 6. Finally, we discuss the limitations with our approach in Sect. 7 before concluding.

2 Related work

To put the proposed work in a broader healthcare perspective, an overview of similar work in mHealth is provided by Sect. 2.1. Next, we discuss the extent to which reinforcement learning/bandit algorithms have been deployed in mHealth settings (Sect. 2.1). IntelligentPooling has similarities with several modeling approaches, here we discuss the most relevant: multi-task learning, meta-learning, Gaussian processes for Thompson Sampling contextual bandits, and time-delayed bandits. These topics are discussed in Sects. 2.2–2.4.

2.1 Connections to Bandit algorithms in mHealth

Bandit algorithms in mHealth have typically used one of two approaches. The first approach is person specific, that is, an algorithm is deployed separately on each user, such as in Rabbi et al. 2015; Jaimes et al. 2016; Forman et al. 2018 and Liao et al. 2020. This approach makes sense when users are highly heterogeneous, that is, their optimal policies differ greatly one from another. However, this approach can present challenges for policy learning 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 (see (Xia 2018) for an empirical evaluation of the shortcomings of Thompson sampling for personalized contextual bandits in mHealth settings). The second approach completely pools users' data, that is one algorithm is used on all users so as to learn a common treatment policy both in bandit algorithms (Paredes et al. 2014; Yom-Tov et al. 2017), and in full reinforcement learning algorithms (Clarke et al. 2017; Zhou et al. 2018). This second approach can potentially learn quickly but may result in poor performance if there is large heterogeneity between users. We compare to these two approaches empirically as they not only represent state-of-the-art in practice, they also represent two intuitive theoretical extremes.

In IntelligentPooling we strike a balance between these two extremes, adjusting the degree of pooling to the degree that users are similarly responsive. When users are heterogeneous, IntelligentPooling achieves lower regret than the second approach while learning more quickly than the first approach. When users are homogeneous our method performs as well as the second approach.

2.2 Connections to multi-task learning and meta-learning

Following original work on non-pooled linear contextual bandits (Agrawal and Goyal 2013), researchers have proposed pooling data in a variety of ways. For example, Deshmukh et al. (2017) proposed pooling data from different arms of a single bandit problem. Li and Kar 2015 used context-sensitive clustering to produce aggregate reward estimates for the bandit algorithm. More relevant to this work is multi-task Gaussian Process (GP), e.g., Lawrence and Platt 2004; Bonilla et al. 2008; Wang and Khardon 2012, however these have been proposed in the prediction as opposed to the reinforcement learning setting. The Gang of Bandits approach (Cesa-Bianchi et al. 2013), which is a generalization from the original LinUCB algorithm for a single task (Li et al. 2010), has been shown to be successful when there is prior knowledge on the similarities between users. For example, a known social network graph might provide a mechanism for pooling. It was later extended to the Horde of Bandits in (Vaswani et al. 2017) which used Thompson Sampling, allowing the algorithm to deal with a large number of tasks.

Each of the multi-task approaches introduces some concept of similarity between users. The extent to which a given user's data contributes to another user's policy is some function of this similarity measure. This is fundamentally different from the approach taken in IntelligentPooling. Rather than determining the extent to which any two users are similar, IntelligentPooling determines the extent to which a given user's reward function parameters differ from parameters in a population (average over all users) reward function. This approach has the advantage of requiring fewer hyper-parameters, as we do not need to learn a similarity function between users. Instead of a pairwise similarity function it is as if we

are learning a similarity between each user and the population average. In the limited data setting, we expect this simpler model to be advantageous.

In meta-learning, one exploits shared structure across tasks to improve performance on new tasks. IntelligentPooling thus shares similarities with meta-learning for reinforcement learning (Nagabandi et al. 2018; Finn et al. 2019; Finn et al. 2018; Zintgraf et al. 2019; Gupta et al. 2018; Sæmundsson et al. 2018). At a high level, one can view our method as a form of meta-learning where the population-level parameters are learned from all available data and each user's parameters represent deviations from the shared parameters. However, while meta-learning might require a large collection of source tasks, we demonstrate the efficacy of our approach on data on the small scale found in clinical mHealth studies.

2.3 Connections to Gaussian process models for Thompson sampling contextual bandits

IntelligentPooling is based on Bayesian mixed effects model of the reward, which is similar to using a Gaussian Process (GP) model with a simple form of the kernel. GP models have been used for multi-armed bandits (Chowdhury and 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) , and for contextual bandits (Li et al. 2010; Krause and Ong 2011). However the above approaches do not structure the way in which the pooling of data across users occurs. IntelligentPooling uses a mixed effects GP model to pool across users in structured manner. Although mixed effects GP models have been previously used for offline data analysis (Shi et al. 2012; Luo et al. 2018), to the best of our knowledge they have not been previously used in the online decision making setting considered in this work.

2.4 Connection to non-stationary linear bandits

There is a growing literature investigating how to adapt linear bandit algorithms to changing environments. A common approach is for the learning algorithm to differentially weight data across time. Differential weighting is used by both Russac et al. 2019 (using a LinUCB algorithm) and Kim and Tewari 2019 (using perturbation-based algorithms). Cheung et al. 2018 to estimate the parameters in the reward function and (Zhao et al. 2020) restart the algorithm at regular intervals discarding the prior data. Similarly (Bogunovic et al. 2016), using GP-based UCB algorithms, accommodate non-stationarity by both restarting and using an autoregressive model for the rewards function. Kim and Tewari 2020 analyze the non-stationary setting with randomized exploration. Wu et al. introduce a model which detects abrupt time changes cite (https://dl.acm.org/doi/pdf/10.1145/3209978.3210051).

IntelligentPooling allows for non-stationary reward functions by the use of time-varying random effects. The correlation between the time-varying random effects induces a weighted estimator whereby more weight is put on the recently collected samples, similar to the discounted estimators in Russac et al. 2019 and Kim and Tewari 2019. In contrast to existing approaches, IntelligentPooling considers both individual and time-specific variation.

3 Intelligent Pooling

IntelligentPooling is a generalization of a Thompson sampling contextual bandit for learning personalized treatment policies. We first outline the components of IntelligentPooling and then introduce the problem definition in Sect. 3.2. As our approach offers a natural alternative to two commonly used approaches, we begin by describing these simpler methods in Sect. 3.3. We introduce our method in Sect. 3.4.

3.1 Overview

The central component of IntelligentPooling is a Bayesian model for the reward function. In particular, IntelligentPooling uses a Gaussian mixed effects linear model for the reward function. Mixed effects models are widely used across the health and behavioral sciences to model the variation in the linear model parameters across users (Raudenbush and Bryk 2002; Laird and Ware 1982) and within a user across time. Use of these models enhances the ability of domain scientists to inform and critique the model used in IntelligentPooling. The properties and pitfalls of these models are well understood; see (Qian et al. 2019) for an application of a mixed effects model in mHealth. IntelligentPooling uses Bayesian inference for the mixed effects model. As discussed in Sect. 2.3, a Bayesian mixed effects linear model is a GP model with a simple kernel. This facilitates increasing the flexibility of the model for the reward function, given sufficient data.

Furthermore, IntelligentPooling uses Thompson sampling (Thompson 1933), also known as posterior sampling (Russo and Van Roy 2014), to select actions. At each decision point, the parameters in the model for the reward function are sampled from their posterior distribution, thus inducing exploration over the action space (Russo et al. 2018). These sampled parameters are then used to form an estimated reward function and the action with the highest estimated reward is selected.

The hyper-parameters (e.g., the variance of the random effects) control the extent of pooling across users and across decision times. The right amount of pooling depends on the heterogeneity among users and the non-stationarity, which is often difficult to pre-specify. Unlike other bandit algorithms in which the hyper-parameters are set at the beginning (Deshmukh et al. 2017; Cesa-Bianchi et al. 2013; Vaswani et al. 2017), IntelligentPooling includes a procedure for updating the hyper-parameters online. In particular, empirical (Bayes Carlin and Louis 2010) is used to update the hyper-parameters in the online setting, as more data becomes available.

3.2 Problem formulation

Consider an mHealth study which will recruit a total of N users.[1] Let $i \in [N] = \{1, \dots, N\}$ be a user index. For each user, we use $k \in \{1, 2, \dots\}$ to index decision times, i.e., times at which a treatment could be provided. Denote by $S_{i,k}$ the states/contexts at the k^{th} decision time of user i. For simplicity, we focus on the case where the action is binary, i.e., $A_{i,k} \in \{0, 1\}$. The algorithm can be easily generalized to cases with more than two actions. After the action $A_{i,k}$ is chosen, the reward $R_{i,k}$ is observed. Throughout the remainder of the

[1] More generally, one can consider the setting where users become known to an algorithm over time. For example, users may open or delete accounts on an online shopping platform.

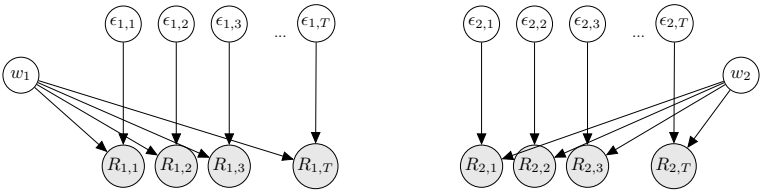

Fig. 1 Consider a setting with two users, here we show the relationship between select random variables in our model: $R_{i,k}$ the reward for user i at decision time k, $\sigma^2_{\epsilon_{i,k}}$ the noise for user i at time k and $\boldsymbol{w}_i$ the latent weight vector for user i. In PERSON-SPECIFIC we see that each user's parameters are independent. Only the prior parameter values are shared, all else is updated independently

paper, S, A and R are random variables and we use lower-case (s, a and r) to refer to a realization of these random variables.

Below we consider a simpler setting where the parameters in the reward are assumed time-stationary. We discuss how to generalize the algorithm to the non-stationary setting in Sect. 6. The goal is to learn personalized treatment policies for each of the N users. We treat this as N contextual bandit problems as the reward function may differ between users. In mHealth settings this might occur due to the inability of sensors to record users' entire contexts. Section 3.3 reviews two approaches for using Thompson Sampling (Agrawal and Goyal 2012) and Sect. 3.4 presents INTELLIGENTPOOLING, our approach for learning the treatment policy for any specific user.

3.3 Two Thompson sampling instantiations

First, consider learning the treatment policy separately per person. We refer to this approach as PERSON-SPECIFIC. At each decision time k, we would like to select a treatment $A_{i,k} \in \{0, 1\}$ based on the context $S_{i,k}$. We model the reward $R_{i,k}$ by a Bayesian linear regression model: for user i and time k

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

where $\phi(s, a)$ is a pre-specified mapping from a context s and treatment a (e.g., those described in Sect. 4.2), w_i is a vector of weights which we will learn, and $\epsilon_{i,k} \sim \mathbf{N}(0, \sigma^2_\epsilon)$ is the error term. The weight vectors $\{w_i\}$ are assumed independent across users and to follow a common prior distribution $w_i \sim \mathbf{N}(\mu_w, \Sigma_w)$. See Fig. 1 for a graphical representation of this approach.

Now at the k^{th} decision time with the context $S_{i,k} = s$, PERSON-SPECIFIC selects the treatment $A_{i,k} = 1$ with probability

$$\pi_{i,k} = \Pr\{\phi(s, 1)^\top \tilde{w}_{i,k} > \phi(s, 0)^\top \tilde{w}_{i,k}\} \tag{2}$$

where $\tilde{w}_{i,k}$ follows the posterior distribution of the parameters w_i in the model (1) given the user's history up to the current decision time k. We emphasize that in this formulation the posterior distribution of w_i is formed based each user's own data.

The opposite approach is to learn a common bandit model for all users. In this approach, the reward model is a single Bayesian regression model with no individual-level parameters:

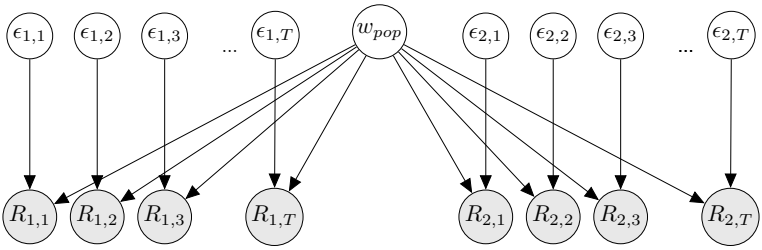

Fig. 2 Consider a setting with two users, here we show the relationship between select random variables in our model: $R_{i,k}$ the reward for user i at decision time k, ϵ_k the noise at time k and w_{pop} the latent weight vector. In COMPLETE we see that each user's parameters are the same. With each parameter update the weight vector for every user is also updated

$$R_{i,k} = \phi(S_{i,k}, A_{i,k})^\top w + \epsilon_{i,k}. \tag{3}$$

where the common parameters, w, follows the prior distribution $w \sim \mathbf{N}(\mu_w, \Sigma_w)$. See Fig. 2 for the graphical representation of this approach. We then use the posterior distribution of the weight vector w to sample treatments for each user. Here the posterior is calculated based on the available data from all users observed up to and including time k. This approach, which we refer to as COMPLETE, may suffer from high bias when there is significant heterogeneity among users.

3.4 Intelligent pooling across bandit problems

INTELLIGENTPOOLING is an alternative to the two approaches mentioned above. Specifically, in INTELLIGENTPOOLING data is pooled across users in an adaptive way, i.e., when there is strong homogeneity observed in the current data, the algorithm will pool more from others than when there is strong heterogeneity.

3.4.1 Model specification

We model the reward associated with taking action $A_{i,k}$ for user i at decision time k by the linear model (1). Unlike PERSON-SPECIFIC where the person-specific weight vectors $\{w_i, i \in [N]\}$ are assumed to be independent to each other, INTELLIGENTPOOLING imposes structure on the w_i's, in particular, a random-effects structure (Raudenbush and Bryk 2002; Laird and Ware 1982):

$$w_i = w_{pop} + u_i, \tag{4}$$

where w_{pop} is a population-level parameter and u_i is a *random effect* that represents the person-specific deviation from w_{pop} for user i. The extent to which the posterior means for w_{pop} and u_i are based on user i's data relative to the population depends on the variances of the random effects (for a stylized example of this see Sect. 3.5). In Sect. 6 we show how we can modify this structure to include time-specific parameters, or a time-specific random effect. A graphical representation for INTELLIGENTPOOLING is shown in Fig. 3.

We assume the prior on w_{pop} is Gaussian with prior mean μ_w and variance Σ_w. u_i is also assumed to be Gaussian with mean $\mathbf{0}$ and covariance Σ_u. Furthermore, we assume $u_i \perp\!\!\!\perp u_j$

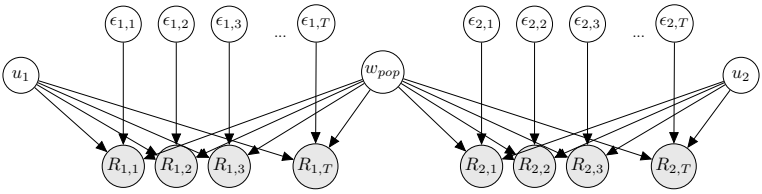

Fig. 3 Consider a setting with two users, here we show the relationship between select random variables in our model: $R_{i,k}$ the reward for user i at decision time k, $\epsilon_{i,k}$ the noise for user i at time k, w_{pop} the latent weight vector and u_i the random effect for user i. In IntelligentPooling we see that some parameters (w_{pop}) are shared across the population which others (u_i) are user specific

for $i \neq j$ and $w_{pop} \perp\!\!\!\perp \{u_i\}$. The prior parameters μ_w, Σ_w as well as the variance of the random effect Σ_u, and the residual variance σ_ϵ^2 are hyper-parameters. In (4), there is a the random effect, u_i on each element of w_i. In practice, one can use domain knowledge to specify which of the parameters should include random effects; this will be the case in the feasibility study described in Sect. 6. Conditioned on the latent variables (w_{pop}, u_i), as well as the current context and action, the expected reward is

$$E[R_{i,k}|w_{pop}, u_i, S_{i,k} = s, A_{i,k} = a] = \phi(s,a)^T(w_{pop} + u_i).$$

3.4.2 Model connections to Gaussian processes

Under the Gaussian assumption on the distribution of the reward and prior, the Bayesian linear model of the reward (1) together with the random effect model (4) can be viewed as an example of Gaussian Process with a special kernel (see Eq. 5). We use this connection to derive the posterior distribution and facilitate the hyper-parameter selection. An additional advantage of viewing the Bayesian mixed effects model as a Gaussian Process model is that we can now flexibly redesign our reward model simply by introducing new kernel functions. Here, we assume linear model with a person-specific random effects. In Sect. 6 we discuss a generalization to time-specific random effects. Additionally, one could adopt non-linear kernels and incorporate more complex structures on the reward function.

3.4.3 Posterior distribution of the weights on the feature vector

In the setting where both the prior and the linear model for the reward follow a Gaussian distribution, the posterior distribution of w_i follows a Gaussian distribution and there are analytic expressions for these updates, as shown in (Williams and Rasmussen 2006). Below we provide the explicit formula of the posterior distribution based on the connection to a Gaussian Process regression. Suppose at the time of updating the posterior distribution, the available data collected from all current users is $\mathcal{D}$, where $\mathcal{D}$ consists of n tuples of state, action, reward and user index $x = (s, a, r, i)$. The mixed effects model (Eqs. 1 and 4) induces a kernel function K. For any two tuples in $\mathcal{D}$, e.g., $x_l = (s_l, a_l, r_l, i_l), l = 1, 2$

$$K(x_1, x_2) = \phi(s_1, a_1)^\top(\Sigma_w + 1_{\{i_1=i_2\}}\Sigma_u)\phi(s_2, a_2). \tag{5}$$

Note that the above kernel depends on Σ_w and Σ_u (one of the hyper-parameters that will be updated using empirical Bayes approach; see below). The kernel matrix $\mathbf{K}$ is of size $n \times n$

and each element is the kernel value between two tuples in $\mathcal{D}$. The posterior mean and variance of w_i given the currently available data $\mathcal{D}$ can be calculated by

$$\begin{aligned} \hat{w}_i &= \mu_w + M_i^\top (\mathbf{K} + \sigma_\epsilon^2 I_n)^{-1} \tilde{R}_n \\ \Sigma_i &= \Sigma_w + \Sigma_u - M_i^\top (\mathbf{K} + \sigma_\epsilon^2 I_n)^{-1} M_i \end{aligned} \tag{6}$$

where $\tilde{R}_n$ is the vector of the rewards centered by the prior means, i.e., each element corresponds to a tuple (s, a, r, j) in $\mathcal{D}$ given by $r - \phi(s,a)^\top \mu_w$, and M_i is a matrix of size n by p (recall p is the length of w_i), with each row corresponding to a tuple (s, a, r, j) in $\mathcal{D}$ given by $\phi(s,a)^\top (\Sigma_w + 1_{\{j=i\}} \Sigma_u)$.

3.4.4 Treatment selection

To select a treatment for user i at the k^{th} decision time, we use the posterior distribution of w_i formed at the most recent update time T. That is, for the context $S_{i,k}$ of user i at the k^{th} decision time, IntelligentPooling selects the treatment $A_{i,k} = 1$ with the probability calculated in the same formula as in (2) but with a different posterior distribution as discussed above.

3.4.5 Setting hyper-parameter values

Recall that the algorithm requires the hyper-parameters μ_w, Σ_w, Σ_u, and σ_ϵ^2. The prior mean μ_w and variance Σ_w of the population parameter w_{pop} can be set according to previous data or domain knowledge (see Sect. 5 for a discussion on how the prior distribution is set in the feasibility study). As we mention in Sect. 3.1, the variance components in the mixed effects model impact how the users pool the data from others (see Sect. 3.5 for a discussion) and might be difficult to pre-specify. IntelligentPooling uses, at the update times, the empirical (Bayes Carlin and Louis 2010) approach to choose/update $\lambda = (\Sigma_u, \sigma_\epsilon^2)$ based on the currently available data. To be more specific, suppose at the time of updating the hyper-parameters, the available data is $\mathcal{D}$. We choose λ to maximize $l(\lambda|\mathcal{D})$, the marginal log-likelihood of the observed reward, marginalized over the population parameters w_{pop} and the random effects u_i. The marginal log-likelihood $l(\lambda|\mathcal{D})$ can be expressed as

$$l(\lambda|\mathcal{D}) = -\frac{1}{2}\left\{ \tilde{R}_n^\top [\mathbf{K}(\lambda) + \sigma_\epsilon^2 I_n]^{-1} \tilde{R}_n + \log\det[\mathbf{K}(\lambda) + \sigma_\epsilon^2 I_n] + n\log(2\pi) \right\} \tag{7}$$

where $\mathbf{K}(\lambda)$ is the kernel matrix as a function of parameters $\lambda = (\Sigma_u, \sigma_\epsilon^2)$. The above optimization can be efficiently solved using existing Gaussian Process regression packages; see Sect. 4.2 for more details.

Algorithm 1 IntelligentPooling

Let $\mathcal{T}$ be a set of all times at which the algorithm might deliver a treatment or perform a parameter update.
`Set` $\hat{w}_{i,0} = \mu_w, \Sigma_{i,0} = \Sigma_w + \Sigma_u$ `for all` i `and` $\mathcal{D} = \{\}$.
for all $t \in \mathcal{T}$ **do**
  if t is a decision time **then**
    `Receive user index` i `and decision time index` k
    `Collect state variable` $S_{i,k}$
    `Calculate randomization probability` $\pi_{i,k} = \Pr_{\tilde{w} \sim \mathbf{N}(\hat{w}_i, \Sigma_i)}\{\phi(S_{i,k}, 1)^\top \tilde{w} > \phi(S_{i,k}, 0)^\top \tilde{w}\}$
    `Sample treatment` $A_{i,k} \sim \text{Bern}(\pi_{i,k})$
    `Collect reward` $R_{i,k}$
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{S_{i,k}, A_{i,k}, R_{i,k}, i\}$
  end if
  if t is an update time **then**
    `Update the hyper-parameters:` $\hat{\lambda} = \text{argmax}\, l(\lambda|\mathcal{D})$ `in Eqn 7`
    `Update the posterior mean and covariance` $\hat{w}_i, \Sigma_i$ `for all` i `in` $\mathcal{D}$ `by Eqns 6 with` $\hat{\lambda}$
  end if
end for

3.5 Intuition for the use of random effects

IntelligentPooling uses random effects to adaptively pool users' data based on the degree to which users exhibit heterogeneous rewards. That is, the person-specific random effect should outweigh the population term if users are highly heterogeneous. If users are highly homogeneous, the person-specific random effect should be outweighed by the population term. The amount of pooling is controlled by the hyper-parameters, e.g., the variance components of the random effects.

To gain intuition, we consider a simple setting where the feature vector ϕ in the reward model (Eq. 1) is one-dimensional (i.e., $p = 1$) and there are only two users (i.e., $i = 1, 2$). Denote the prior distributions of population parameter w_{pop} by $\mathbf{N}(0, \sigma_w^2)$ and the random effect u_i by $\mathbf{N}(0, \sigma_u^2)$. Below we investigate how the hyper-parameter (e.g., σ_u^2 in this simple case) impacts the posterior distribution.

Let k_i be the number of decision time of user i at an updating time. In this simple setting, the posterior mean of $\hat{w}_1$ can be calculated explicitly:

$$\hat{w}_1 = \frac{[\delta\gamma + (1-\gamma^2)C_2]Y_1 + \delta\gamma^2 Y_2}{(1-\gamma^2)C_1C_2 + \delta\gamma(C_1 + C_2) + (\delta\gamma)^2}$$

where for $i = 1, 2$, $C_i = \sum_{k=1}^{k_i} \phi(A_{i,k}, S_{i,k})^2$, $Y_i = \sum_{k=1}^{k_i} \phi(A_{i,k}, S_{i,k}) R_{i,k}$, $\gamma = \sigma_w^2/(\sigma_w^2 + \sigma_u^2)$ and $\delta = \sigma_\epsilon^2/\sigma_w^2$. Similarly, the posterior mean of w_2 is given by

$$\hat{w}_2 = \frac{[\delta\gamma + (1-\gamma^2)C_1]Y_2 + \delta\gamma^2 Y_1}{(1-\gamma^2)C_1C_2 + \delta\gamma(C_1 + C_2) + (\delta\gamma)^2}$$

When $\sigma_u^2 \to 0$ (i.e., the variance of random effect goes to 0), we have $\gamma \to 1$ and both posterior means $(\hat{w}_1, \hat{w}_2)$ approach the posterior mean under Complete (Eqn 3) using prior $\mathbf{N}(0, \sigma_w^2)$

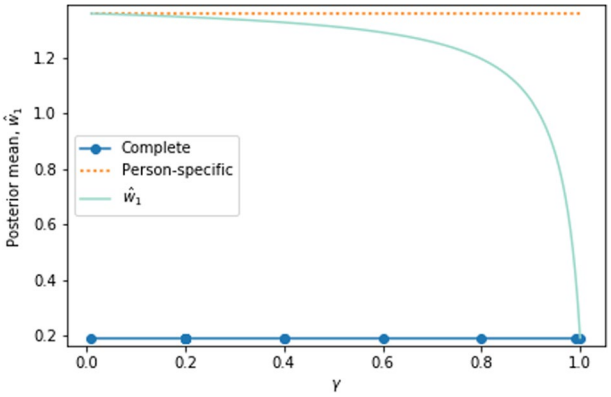

Fig. 4 The posterior mean of w_i, $\hat{w}_1$. As the variance of random effect σ_u^2 decreases, γ increases and the posterior mean approaches the population-informed estimation (COMPLETE) and departs from the person-specific estimation (PERSON-SPECIFIC).

$$\hat{w}_1, \hat{w}_2 \to \frac{Y_1 + Y_2}{C_1 + C_2 + \delta}.$$

Alternatively, when $\sigma_u^2 \to \infty$, we have $\gamma \to 0$ and the posterior means $(\hat{w}_1, \hat{w}_2)$ each approach their respective posterior means under PERSON-SPECIFIC (Eqn 1) using a non-informative prior

$$\hat{w}_1 \to \frac{Y_1}{C_1}, \ \hat{w}_2 \to \frac{Y_2}{C_2}.$$

Figure 4 illustrates that when γ goes from 0 to 1, the posterior mean $\hat{w}_i$ smoothly transitions from the population estimates to the person-specific estimates.

3.6 Regret

We prove a regret bound for a modification of INTELLIGENTPOOLING similar to that in Agrawal and Goyal 2012; Vaswani et al. 2017 in a simplified setting. Further details are provided in Appendix 1. Let d be the length of the weight vector w_i in the Bayesian mixed effects model of the reward in Eq. 1. Recall that Σ_w is the prior covariance of the weight vector w_{pop}, Σ_u is the covariance of the random effect u_i and σ_ϵ^2 is the variance of the error term. Let K_i be the number of decision times for user i up to a given calendar time and $T = \sum_{i=1}^{N} K_i$ be the total number of decision times encountered by all N users in the study up to the calendar time. We define the regret of the algorithm after T decision times by $\mathcal{R}(T) = \sum_{i=1}^{N} \sum_{k=1}^{K_i} \max_a \phi(S_{i,k}, a)^T w_i - \phi(S_{i,k}, A_{i,k})^T w_i$.

Theorem 1 *With probability* $1 - \delta$, *where* $\delta \in (0, 1)$ *the total regret of the modified Thompson Sampling with* INTELLIGENTPOOLING *after T total number of decision times is:*

$$R(T) = \tilde{O}\left(dN\sqrt{T} \sqrt{\log\left(\frac{(\mathrm{Tr}(\Sigma_w) + \mathrm{Tr}(\Sigma_u) + \mathrm{Tr}(\Sigma_u^{-1}))}{d} + \frac{T}{\sigma_\epsilon^2 dN} \right) \log \frac{1}{\delta}} \right)$$

Remark Observe that, up to logarithmic terms, this regret bound is $\tilde{O}(dN\sqrt{T})$. Recall that (Vaswani et al. 2017) introduces a similar regret bound for a Thompson Sampling algorithm which utilizes user-similarity information. The bound from (Vaswani et al. 2017),

$\tilde{O}(dN\sqrt{T/\lambda})$, additionally depends on a hyper-parameter λ that is not included in our model. In (Vaswani et al. 2017), λ controls the strength of prior user-similarity information. Instead of introducing a hyper-parameter our model follows a mixed effects Bayesian structure which allows user similarities (as expressed in the extent to which users' data is pooled) to be updated with new data. Thus, in certain regimes of hyper-parameter λ, IntelligentPooling will incur much smaller regret, as demonstrated empirically in Sect. 4.3.

4 Experiments

This work was conducted to prepare for deployment of IntelligentPooling in a live trial. Thus, to evaluate IntelligentPooling we construct a simulation environment from a precursor trial, HeartStepsV1 (Klasnja et al. 2015). This simulation allows us to evaluate the proposed algorithm under various settings that may arise in implementation. For example, heterogeneity in the observed rewards may be due to unknown subgroups across which users' reward functions differ. Alternatively, this heterogeneity may vary across users in a more continuous manner. We consider both scenarios in simulated trials. In Sects. 4.1-4.3 we evaluate the performance of IntelligentPooling against baselines and a state-of-the-art algorithm. In Sect. 5 we assess feasibility of IntelligentPooling in a pilot deployment in a clinical trial.

4.1 Simulation environment

HeartStepsV1 was a 6-week micro-randomized trial of an Android-based physical activity intervention with 41 sedentary adults. The intervention consisted of two *push* interventions: planning and contextually-tailored activity suggestions. Activity suggestions acted as action cues and were designed to provide users with actionable options for engaging in short bouts of activity in their current situation. The content of the suggestions was tailored based on the users' location, weather, time of day, and day of the week. For each individual, on each day of the study, the HeartSteps system randomized whether or not to send an activity suggestion five times a day. The intended outcome of the suggestions—the proximal outcome used to evaluate their efficacy—was the step count in the 30 minutes following suggestion randomization.

HeartStepsV1 data was used to construct all features within the environment, and to guide choices such as how often to update the feature values. Recall that $S_{i,k}$ and $R_{i,k}$ denote the context features and reward of user i at the k^{th} decision time. The reward is the log step counts in the thirty minutes immediately following a decision time. In HeartStepsV1 three treatment actions were considered: $A_{i,k} = 1$ corresponded to a smartphone notification containing an activity suggestion designed to take 3 minutes to perform, $A_{i,k} = 0$ corresponded to a smartphone notification containing an anti-sedentary message designed to take approximately 30 seconds to perform and $A_{i,k} = -1$ corresponded to not sending a message. However, in the simulation only the actions 1, 0 are considered.

Figure 5 describes the simulation while Table 1 describes context features and rewards. Each context feature in Table 1 was constructed from HeartStepsV1 data. For example, we found that in HeartStepsV1 data splitting participants' prior 30 minute step count into the two categories of high or low best explained the reward. Additional details about this process are included in Appendix 4.

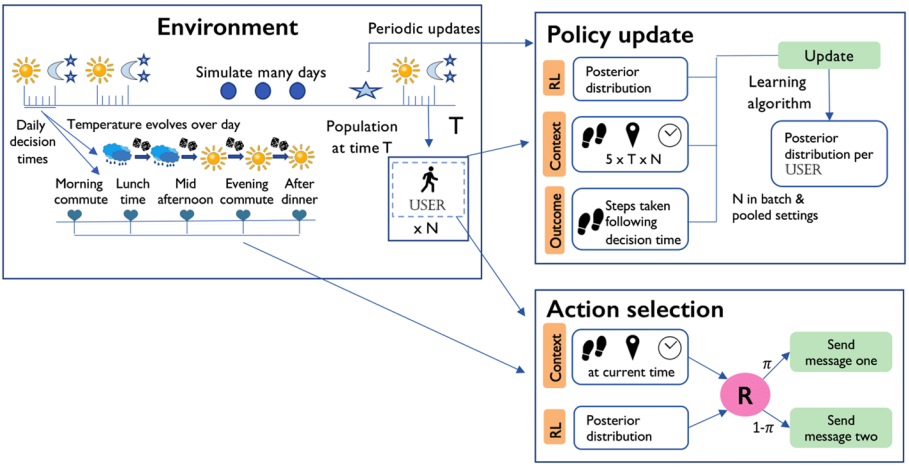

Fig. 5 Contextual features for a simulated User are composed of both general environmental features (such as time of day) and individual features (such as location). At decision times a simulated user receives a message determined by the current treatment policy. Periodically this policy is updated according to a learning algorithm which outputs a new posterior distribution for each User

Table 1 The value used in encoding each feature is shown in parentheses

Name	Value	User Specific
Time of day	Morning 9:00 and 15:00 (0) Afternoon 15:00 and 21:00 (1)	No
Day of the week	Weekday (0) or Weekend (1)	No
Temperature	Cold (0) or Hot (1)	No
Preceding activity level	Low (0) or High (1)	Yes
Location	Other (0) or Home/work (1)	Yes
Intercept	1	Yes
Reward		
Step count	Continuous on log scale	Yes

For example cold (0) indicates that cold is coded as a 0 wherever this feature is used. A user's state is described as $S_{i,k} = \{1, \text{time of day}, \text{day of the week}, \text{preceding activity level}, \text{location}\}$

The temperature and location are updated throughout a simulated day according to probabilistic transition functions constructed from HeartStepsV1. The step counts for a simulated user are generated from participants in HeartStepsV1 as follows. We construct a one-hot feature vector containing the group-ID of a participant, the time of day, the day of the week, the temperature, the preceding activity level, and the location. Then for each possible realization of the one-hot encoding we calculate the empirical mean and empirical standard deviation of all step counts observed in HeartStepsV1. The corresponding empirical mean and empirical standard deviation from HeartStepsV1 form $\mu_{S_{i,k}}$ $\sigma_{S_{i,k}}$ respectively. At each 30 minute window, if a treatment is not delivered step counts are generated according to

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

Homogeneous	Bi-modal	Smooth
$Z^i = 0\ \beta_i^l{=}0$	$Z_i, \beta_i^l = \begin{cases} z_1, \beta_1^l & \text{if } i \in \text{group one} \\ z_2, \beta_2^l & \text{if } i \in \text{group two} \end{cases}$	$Z_i \sim \mathcal{N}(0, \sigma^2)\ \beta_i^l \sim \mathcal{N}(0, \sigma_l^2)$

$$R_{i,k} = \mathbf{N}(\mu_{S_{i,k}}, \sigma^2_{S_{i,k}}). \tag{8}$$

Heterogeneity This model, which we denote HETEROGENEITY, allows us to compare the performance of the approaches under different levels of population heterogeneity. The step count after a decision time is a modification of Eq. 8 to reflect the interaction between context and treatment on the reward and heterogeneity in treatment effect. Let β be a vector of coefficients of $S_{i,k}$ which weigh the relative contributions of the entries of $S_{i,k}$ that interact with treatment on the reward. The magnitude of the entries of β are set using HEARTSTEPSV1. Step counts ($R_{i,k}$) are generated as

$$R_{i,k} = \mathbf{N}(\mu_{S_{i,k}}, \sigma^2_{S_{i,k}}) + A_{i,k}(S_{i,k}^T\beta_i + Z_i). \tag{9}$$

The inclusion of Z_i will allow us to evaluate the relative performance of each approach under different levels of population heterogeneity. Let β_i^l be the entry in β_i corresponding to the location term for the i^{th} user. We consider three scenarios (shown in Table 6) to generate Z_i, the person-specific effect, and β_i^l the location-dependent effect. The performance of each algorithm under each scenario will be analyzed in Sect. 4.3. In the smooth scenario, σ is equal to the standard deviation of the observed treatment effects $[f(S_{i,k})^\top\beta\ :\ S_{i,k} \in \text{HEARTSTEPSV1}]$. The settings for all Z_i and β_i^l terms are discussed in Sect. D.

In the bi-modal scenario each simulated user is assigned a base-activity level: low-activity users (group 1) or high-activity users (group 2). When a simulated user joins the trial they are placed into either group one or two with equal probability. Whether or not it is optimal to send a treatment (an activity suggestion) for user i at their k^{th} decision time depends both on their context, and on the values of z_1, β_1^l and z_2, β_2^l. The values of z_1, β_1^l and z_2, β_2^l are set so that for all users in group 1, it is optimal to send a treatment under 75% of the contexts they will experience. Yet for all users in group 2, it is only optimal to send a treatment under 25% of the contexts they will experience. Group membership is not known to any of the algorithms Table 2. The settings for all values in Table 6 are included in Sect. D.

4.2 Model for the reward function in INTELLIGENTPOOLING

In Sect. 3 we introduced the feature vector $\phi(S_{i,k}, A_{i,k}) \in \mathbb{R}^p$. This vector is used in the model for the reward and transforms a user's contextual state variables $S_{i,k}$ and the action $A_{i,k}$ as follows:

$$\phi(S_{i,k}, A_{i,k})^T = \left(S_{i,k}^T, \pi_{i,k}S_{i,k}^T, (A_{i,k} - \pi_{i,k})S_{i,k}\right), \tag{10}$$

where $S_{i,k} = \{1, \text{time of day, day of the week, preceding activity level, location}\}$. Recall that the bandit algorithms produce $\pi_{i,k}$ which is the probability that $A_{i,k} = 1$. The inclusion of

the term $(A_{i,k} - \pi_{i,k})S_{i,k}$ 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). In HeartStepsV1 we observed that users varied in their overall responsivity and that a user's location was related to their responsivity. In the simulation, we assume the person-specific random effect on four parameters in the reward model (i.e., the coefficients of terms in S involving the intercept and location).

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 and we implement GP regression with the software package (GPytorch Gardner et al. 2018).

4.3 Simulation results

In this section, we compare the use of mixed effects model for the reward function in IntelligentPooling to two standard methods used in mHealth, Complete and Person − Specific from Sect. 3.3. Recall that IntelligentPooling includes person-specific random effects, as described in Eq. 14. In Person − Specific, all users are assumed to be different and there is no pooling of data and in Complete, we treat all users the same and learn one set of parameters across the entire population.

Additionally, to assess IntelligentPooling's ability to pool across users we compare our approach to Gang of Bandits (Cesa-Bianchi et al. 2013), which we refer to as GangOB. As this model requires a relational graph between users, we construct a graph using the generative model (9) and Table 6 connecting users according to each of the three settings: homogeneous, bi-modal and smooth. For example, with knowledge of the generative model users can be connected to other users as a function of their Z_i terms. As we will not have true access to the underlying generative model in a real-life setting we distort the true graph to reflect this incomplete knowledge. That is we add ties to dissimilar users at 50% of the strength of the ties between similar users.

From the generative model (9), the optimal action for user i at the k^{th} decision time is $a^*_{i,k} = \mathbb{1}_{\{S^T_{i,k}\beta^*_i + Z_i \geq 0\}}$. The regret is

$$\text{regret}_{i,k} = |S^T_{i,k}\beta^*_i + Z_i| \mathbb{1}_{\{a^*_{i,k} \neq A_{i,k}\}} \tag{11}$$

where β^*_i is the optimal β for the *ith* user.

In these simulations each trial has 32 users. Each user remains in the trial for 10 weeks and the entire length of the trial is 15 weeks, where the last cohort joins in week six. The number of users who join each week is a function of the recruitment rate observed in HeartStepsV1. In all settings we run 50 simulated trials.

First, Fig. 6 provides the regret averaged across all users across 50 simulated trials where the reward distribution follows (9) for each of the Table 6 categories. The horizontal axis in Fig. 6 is the average regret over all users in their nth week in the trial, e.g. in their first week, their second week, etc. In the bi-modal setting there are two groups, where all users in group one have a positive response to treatment when experiencing their typical context, while the users in group two have a negative response to treatment under their typical context. An optimal policy would learn to *not typically* send treatments to users in the first group, and to *typically* send them to users in the second. To evaluate each algorithm's

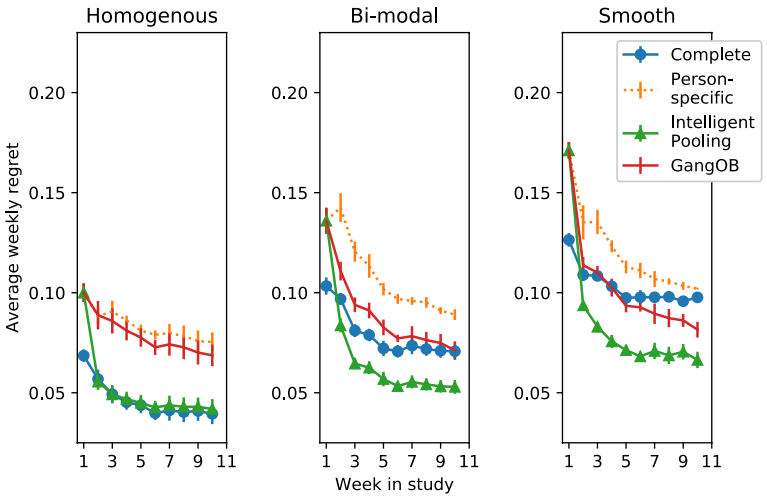

Fig. 6 Heterogeneity generative model Regret averaged across all users for each week in the trial, i.e. average regret of all users in their first week of the trial

Table 3 The fraction of time that messages were sent to users in each group

	Group one optimal policy = send activity suggestion	Group two optimal policy = send anti-sedentary message
Complete	0.49	0.46
Person-specific	0.65	0.49
GangOB	0.57	0.35
Intelligent-Pooling	0.59	0.36

Recall at each decision time either an activity suggestion or anti-sedentary message is sent. For group one it is typically optimal to send an activity suggestion, while for group two it is typically optimal to send an anti-sedentary message. Here, IntelligentPooling is best able to learn this dynamic

ability to learn this distinction we show the percentage of time each group received a message in Table 3.

The relative performance of the approaches depends on the heterogeneity of the population. When the population is very homogenous Complete excels, while its performance suffers as heterogeneity increases. Person-Specific is able to personalize; as shown by Table 3, it can differentiate between individuals. However, it learns slowly and can only approach the performance of Complete in the smooth setting of Table 6 where users differ the most in their response to treatment. Both IntelligentPooling and GangOB are more adaptive than either Complete or Person-Specific. GangOB consistently outperforms Person-Specific and achieves lower regret than Complete in some settings. In the homeogenous setting we see that GangOB can utilize social information more effectively than Person-Specific does while in the smooth setting it can adapt to individual differences more effectively than Complete. Yet, IntelligentPooling demonstrates stronger and swifter adaptability than does GangOB, consistently achieving lower regret at quicker rates. Finally, the algorithms differ in their suitability for real-world applications, especially when data is limited.

GangOB requires reliable values for hyper-parameters and can depend on fixed knowledge about relationships between users. IntelligentPooling can learn how to pool between individuals over time and without prior knowledge.

5 IntelligentPooling feasibility study

The simulated experiments provide insights into the potential of this approach for a live deployment. As we see reasonable performance in the simulated setting, we now discuss an initial pilot deployment of IntelligentPooling in a real-life physical activity clinical trial.

5.1 Feasibility study design

The feasibility study of IntelligentPooling involves 10 participants added to a larger 90-day clinical trial of HeartSteps v2, an mHealth physical activity intervention. The purpose of the larger clinical trial is to optimize the intervention for individuals with Stage 1 hypertension. Study participants with Stage 1 hypertension were recruited from Kaiser Permanente Washington in Seattle, Washington. The study was approved by the institutional review board of the Kaiser Permanente Washington Health Research Institute (under number 1257484-14).

HeartSteps v2 is a cross-platform mHealth application that incorporates several intervention components, including weekly activity goals, feedback on goal progress, planning, motivational messages, prompts to interrupt sedentary behavior, and—most relevant to this paper—actionable, contextually-tailored suggestions for individuals to perform a short physical activity (suggesting, roughly, a 3 to 5 minute walk). In this study physical activity is tracked with a commercial wristband tracker, the Fitbit Versa smart watch.

In this version of the intervention, activity suggestions are randomized five times per day for each participant on each day of the 90-day trial. These decision times are specified by each user at the start of the study, and they roughly correspond to the participant's typical morning commute, lunch time, mid-afternoon, evening commute, and after dinner periods. The treatment options for activity suggestions are binary: at a decision time, the system can either send or not send a notification with an activity suggestion. When provided, the content of the suggestion is tailored to current sensor data (location, weather, time of day, and day of the week). Examples of these suggestions are provided in Klasnja et al. 2018. At a decision time, activity suggestions are randomized only if the system considers that the user is available for the intervention—i.e., that it is appropriate to intervene at that time (see Fig. 8 for criteria used to determine if it is appropriate to send an activity suggestion at a decision time). Subject to these availability criteria, IntelligentPooling determines whether to send a suggestion at each decision time. The posterior distribution was updated once per day, prior to the beginning of each day. Figure 7 provides a schematic of the feasibility study.

The feasibility study included the second set of 10 participants in the trial of HeartSteps v2, following the initial 10 enrolled participants. IntelligentPooling (Algorithm 1) is deployed for each of the second set of 10 participants. At each decision time for these 10 participants, IntelligentPooling uses all data up to that decision time (i.e. from the initial ten participants as well as from the subsequent ten participants). Thus the feasibility study allows us to assess performance of IntelligentPooling after the beginning of a study instead of the performance at the beginning of the study (when there is little data) or the

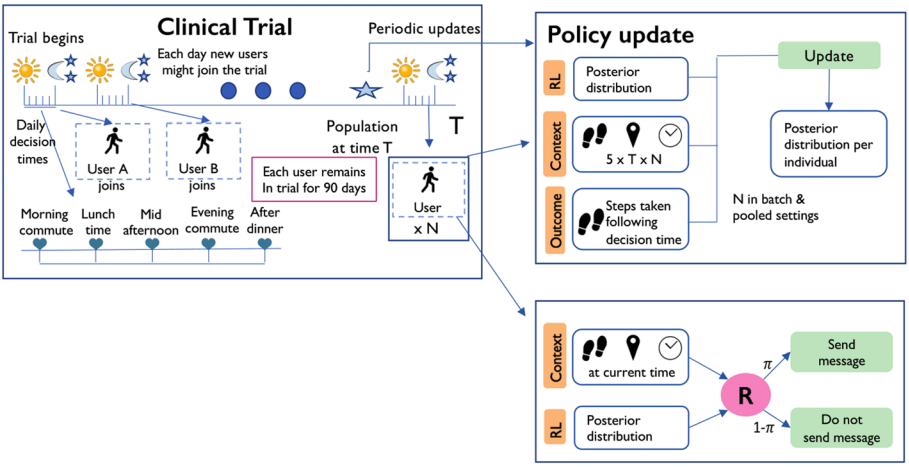

Fig. 7 Setup of FeasibilityStudy. Users can receive treatments up to five times a day during the 90 days. Users enter the trial asynchronously

A user is available to receive an activity suggestion under the following conditions:

- She is not currently active and has not had a large amount of activity in the last two hours.
- She has not recently received a notification with a HeartSteps intervention.
- Her phone has an internet connection and can communicate with the HeartSteps server.
- Her smart watch has been able to communicate with the HeartSteps server in the last ten minutes to provide the current location and step count data.

Fig. 8 Availability criteria

performance at the end of the study (when there is a large amount of data and the algorithm can be expected to perform well).

In the feasibility study, the features used in the reward model were selected to be predictive of the baseline reward and/or the treatment effect, based on the data analysis of HeartStepsV1; see Sect. 6.2 in (Liao et al. 2020) for details. All features used in the reward model are shown in Table 4. The feature *engagement* represents the extent to which a user engages with the mHealth application measured as a function of how many screen views are made within the application within a day. The feature *dosage* represents the extent to which a user has received treatments (activity suggestions). This feature increases and decreases depending on the number of activity suggestions recently received. The feature *location* refers to whether a user is at home or work (encoded as a 1) or somewhere else (encoded as a 0). The *temperature* feature value is set according to the temperature at a user's current location (based off of phone GPS). The *variation* feature value is set according to the variation in step count in the hour around that decision point over the prior seven-day period. As before we construct a feature vector ϕ, however here we only use select terms to estimate the treatment effect. Here,

$$\phi(S_{i,k}, A_{i,k})^T = \left(S_{i,k}^T, \pi_{i,k} S_{i,k}'^T, (A_{i,k} - \pi_{i,k}) S_{i,k}'\right), \tag{12}$$

Table 4 State feature descriptions for FeasibilityStudy

State Features			
Name	Value	User specific	Included in treatment effect
Temperature	Continuous	Yes	No
Yesterday's step count	Continuous	Yes	No
Prior 30-minute step count	Continuous	Yes	No
Step variation level	Discrete	Yes	Yes
Engagement with mobile application	Discrete	Yes	Yes
Dosage	Continuous	Yes	Yes
Location	Discrete	Yes	Yes
Intercept	1	Yes	Yes
Reward			
Step count	Continuous on log scale	Yes	NA

where $S_{i,k} = \{1, \text{temperature, yesterday's step count, preceding activity level, step variation, engagement, dosage, location}\}$ and $S'_{i,k} = \{1, \text{step variation, engagement, dosage, location}\}$ is a subset of $S_{i,k}$.

We provide a full description of these features in Sect. E. The prior distribution was also constructed based on HeartStepsV1; see Sect. 6.3 in (Liao et al. 2020) for more details. As this feasibility study only includes a small number of users, a simple model with only two person-specific random effects, each on the intercept term in S and S' (Eq. 12) was deployed.

Here we discuss how much data we have to personalize the policy to each user. Recall the 10 users only receive interventions when they meet the availability criteria outlined in Fig. 8, thus we find that in practice we have a limited number of decision points to learn a personalized policy from. In the case of perfect availability, we would have at most 450 decision points per person. However due to the criteria in Fig. 8, the algorithm is used with only approximately 23% of each user's decision points. Pooling users' data allows us to learn more rapidly. On the day that the first pooled user joined the feasibility study there were 107 data points from the first set of 10 users.

The 10 users received an average number of .20 ($\pm$0.015) messages a day. The average log step count in the 30-minute window after a suggestion was sent was 4.47, while it was 3.65 in the 30-minute windows after suggestions were not sent. Figure 9 shows the entire history of treatment selection probabilities for all of the users who received treatment according to IntelligentPooling. We see that the treatment probabilities tended to be low, though they covered the whole range of possible values.

We would like to assess the ability of IntelligentPooling to personalize and learn quickly. To do so we perform an analysis of the learning algorithms of IntelligentPooling, Complete and Person-Specific on batch data containing tuples of (S, A, R). Note that the actions in this batch data were selected by IntelligentPooling, however, here we are not interested in the action selection components of each algorithm but instead on their ability to learn the posterior distribution of the weights on the feature vector.

Personalization By comparing how the decisions to treat under IntelligentPooling differ from those under Complete, we gather preliminary evidence concerning whether

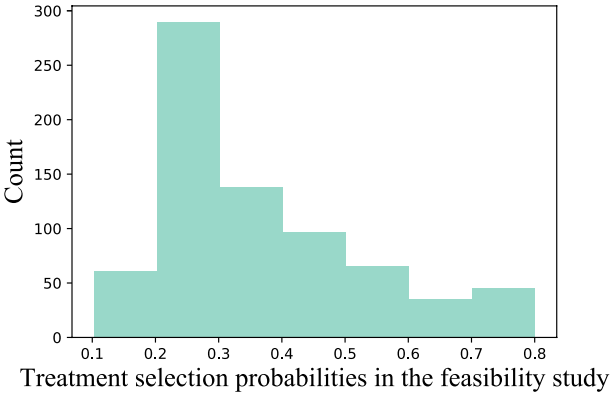

Fig. 9 We see that IntelligentPooling covers the full range of treatment selection probabilities. The tendency seems to be to send with a lower rather than higher probability

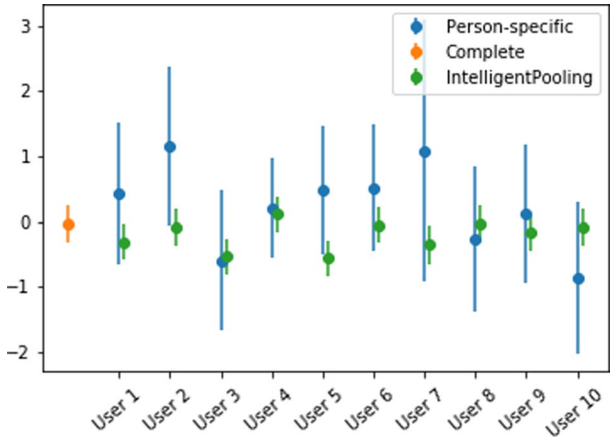

Fig. 10 Posterior mean and standard deviation of the coefficient of $A_{i,k}$ in Eq. 12 for all users in the feasibility study

IntelligentPooling personalizes to users. Figure 10 shows the posterior mean of the coefficient of the $A_{i,k}$ term in the estimation of the treatment effect, for all users in the feasibility study on the 90th day after the last user joined the study. We show this term not only for IntelligentPooling but also for Complete and Person-Specific. We see that for some users this coefficient is below zero while for others it is above. While the terms under IntelligentPooling differ from Complete they do not vary as much as those learned by Person-Specific. Yet, crucially, the variance is much lower for these terms.

Figure 11 displays the posterior mean of the coefficient of the $A_{i,k}$ term in the estimation of the treatment effect. This coefficient represents the overall effect of treatment on one of the users, *User A*. During the prior 7 days User A had not experienced much variation in activity at this time and the user's engagement is low. Note that the treatment appears to have a positive effect on a different user, User B, in this context whereas on User A there is little evidence of a positive effect. If Complete had been used to determine treatment, User A might have been over-treated.

Speed of policy learning We consider the speed at which IntelligentPooling diverges from the prior, relative to the speed of divergence for Person-Specific. Figure 12 provides the Euclidean distance between the learned posterior and prior parameter vectors (averaged across the data from the 10 users at each time). From Fig. 12 we see that Person-Specific

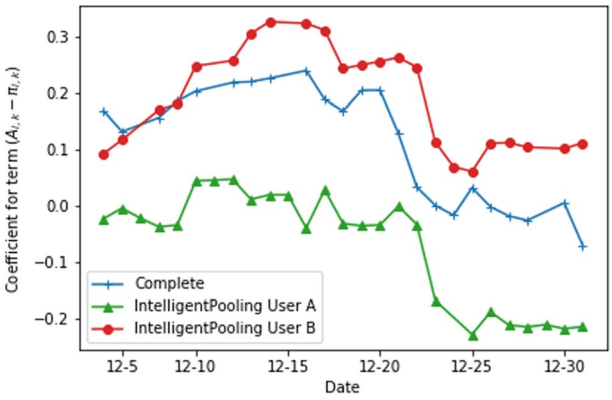

Fig. 11 Posterior mean of the coefficient of $A_{i,k}$ in Eq. 12 for users A and B in the feasibility study

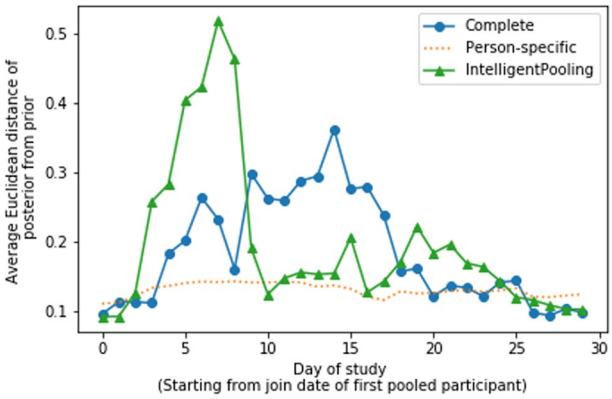

Fig. 12 Mean squared distance of the posterior mean from prior mean of the coefficients of $A_{i,k}$

hardly varies over time in contrast to IntelligentPooling and Complete, which suggests that Person-Specific learns more slowly.

In conclusion IntelligentPooling was found to be feasible in this study. In particular the algorithm was operationally stable within the computational environment of the study, produced decision probabilities in a timely manner, and did not adversely impact the functioning of the overall mHealth intervention application. Overall, IntelligentPooling produced treatment selection probabilities which covered the full range of available probabilities, though treatments tended to be sent with a low probability.

6 Non-stationary environments

An additional challenge in mHealth settings is that users' response to treatment can vary over time. To address this challenge we show that our underlying model can be extended to include time-varying random effects. This allows each policy to be aware of how a user's response to treatment might vary over time. We propose a new simulation to evaluate this approach and show that IntelligentPooling achieves state-of-the-art regret, adjusting to non-stationarity even as user populations vary from heterogenous to homogenous.

6.1 Time-varying random effect

In addition to user-specific random effects we extend our model to include time-specific random effects. Consider the Bayesian mixed effects model with person-specific and time-varying effects: for user i at the k^{th} decision time,

$$R_{i,k} = \phi(S_{i,k}, A_{i,k})^\top w_{i,k} + \epsilon_{i,k}. \tag{13}$$

In addition, we impose the following additive structure on the parameters $w_{i,k}$:

$$w_{i,k} = w_{pop} + u_i + v_k, \tag{14}$$

where w_{pop} is the population-level parameter, u_i represents the person-specific deviation from w_{pop} for user i and v_k is the time-varying random effects allowing $w_{i,k}$ to vary with time in the study.

The prior terms for this model are as introduced in Sect. 3.4. Additionally, v_k has mean $\mathbf{0}$ and covariance D_v. The covariance between two relative decision times in the trial is $\text{Cov}(v_k, v_{k'}) = \rho(k, k')D_v$, where $\rho(k, k') = \exp(-dist(k, k')^2/\sigma_\rho)$ for a distance function, *dist* and $\theta_{pop} \perp\!\!\!\perp \{u_i\}\{v_k\}$. There is no change to Algorithm 1 except that now the algorithm would select the action based on the posterior distribution of $w_{i,k}$, which depends on both the user and time in the study.

6.2 Experiments

We now modify our original simulation environment so that users' responses will vary over time. To do so we introduce the generative model Disengagement. This generative model captures the phenomenon of disengagement. That is as users are increasingly exposed to treatment over time they can become less responsive. This model adds a further term to (9), $A_{i,k}X_w^T\beta_w$ where X_w is defined as follows. Let $w_{i,k}$ be the highest number of weeks user i has completed at time k; X_w encodes a user's current week in a trial, $X_w = [\mathbb{1}_{\{w_{i,k}=0\}}, \ldots, \mathbb{1}_{\{w_{i,k}=11\}}]$. We set β_w such that the longer a user has been in treatment, the less they respond to a treatment message. When a simulated user is at a decision time the user will receive a treatment message according to whichever RL policy is being run through the simulation.

In order to evaluate the effectiveness of our time-varying model we compare to Time-Varying Gaussian Process Thompson Sampling (TV-GP) (Bogunovic et al. 2016). This approach incorporates temporal information for non-stationary environments and was shown to be competitive to stationary models. To compare this method to IntelligentPooling we use a linear kernel for the spatial component. We then modify Eq. 6 to compute the posterior distribution by removing the random-effects and modifying the kernel (Eq. 5) to include the temporal terms introduced in (Bogunovic et al. 2016).

Figure 13 provides the regret averaged across all users across 50 simulated trials where the reward distribution follows generative model Disengagement. As before the horizontal axis in Fig. 13 is the average regret over all users in their n^{th}week in the trial, e.g. in their first week, their second week, etc. In Disengagement, the time-specific response to treatment is set so that a negative response to treatment is introduced in the seventh week of the trial.

In the Disengagement condition as users become increasingly less responsive to treatment good policies should learn to treat less. Thus, Table 5 provides the average number

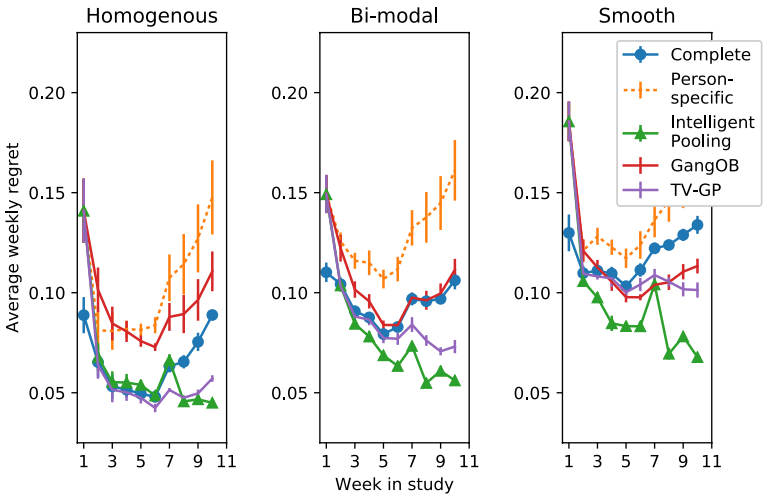

Fig. 13 Disengagement generative model Regret averaged across all users for each week in the trial, i.e. average regret of all users in their first week of the trial

Table 5 Average fraction of times treatment was sent (action=1), over 50 simulations (generative model HETEROGENEITY with homogenous Z^h setting)

	Cohort one week 10	Cohort six week 10
Complete	0.62	0.44
Person-specific	0.76	0.59
HordeOB-	0.50	0.57
TV-GP	0.64	0.31
Intelligent-Pooling	0.30	0.06

of times a treatment is sent in the last week of the trial for both the first and last cohort. We expect that a policy which learns not to treat will treat less often in the last week of the last cohort than in the last week of the first cohort.

7 Limitations

A significant limitation with this work is that our pilot study involved a small number of participants. Our results from this work must be considered with caution as preliminary evidence towards the feasibility of deploying INTELLIGENTPOOLING, and bandit algorithms in general, in mHealth settings. Moreover, we cannot claim to provide generalizable evidence that this algorithm can improve health outcomes; for this larger studies with more participants must be run. We offer our findings as motivation for such future work.

Our proposed model is designed to overcome the challenges faced when learning personalized policies in limited data settings. As such, if data was abundant our model would likely have limited effectiveness compared to more complex models. For example, a more complex model could allow us to pool between users as a function of their similarity. Our

current model instead determines the extent to which a given user deviates from the population and does not consider between-user similarities. A limitation with our current understanding of mHealth is that it is unclear what a good similarity measure would be. We leave the question of designing a data-efficient algorithm for learning such a measure as future work.

A component of IntelligentPooling is the use of empirical Bayes to update the model hyper-parameters. Here, we used an approximate procedure. However, with our model it is possible to produce exact updates in a streaming fashion and we are currently developing such an approach.

Ideally, we would evaluate IntelligentPooling against all other approaches in a clinical trial setting. However, here we only demonstrated the feasibility of our approach on a limited number of users and did not have the resources to similarly test the other approaches. To overcome this limitation we constructed a realistic simulation environment so that we could evaluate on different populations without the costly investment of designing multiple arms of a real-life trial. While the simulated experiments and the feasibility study together demonstrate the practicality of our approach, in future work one might deploy all potential approaches in simultaneous live trials.

Finally, IntelligentPooling can incorporate a time-specific random effect to capture the phenomenon of responsivity changing over the course of a study. There is much to be improved with this model. For example, the first cohort in a study will not have prior cohorts to learn from, and the final cohort will have the greatest amount of data to benefit from. Other models might treat different cohorts with greater equality. Furthermore, this representation does not incorporate alternative temporal information, such as continually shifting weather patterns, where temperatures might change slowly and gradually alter one's desire to exercise outside.

8 Conclusion

When data on individuals is limited a natural tension exists between personalizing (a choice which can introduce variance) and pooling (a choice which can introduce bias). In this work we have introduced a novel algorithm for personalized reinforcement learning, IntelligentPooling that presents a principled mechanism for balancing this tension. We demonstrate the practicality of our approach in the setting of mHealth. In simulation we achieve improvements of 26% over a state-of-the-art-method, while in a live clinical trial we show that our approach shows promise of personalization on even a limited number of users. We view adaptive pooling as a first step in addressing the trade-offs between personalization and pooling. The question of how to quantify the benefits and risks for individual users is an open direction for future work.

Appendix 1: Regret bound

In this section we prove a high probability regret bound for a modification of IntelligentPooling in a simplified setting. We modify the Thompson sampling algorithm in IntelligentPooling by multiplying the posterior covariance by a tuning parameter, following Agrawal and Goyal (2012). This is mainly due to the technical reasons; see Abeille and Lazaric (2017) for a discussion. We also simplify the setting in this regret analysis.

Specifically, we assume that the posterior distribution of all users is updated after every decision time and the hyper-parameters are fixed throughout the study.

Vaswani et al. (2017) also provided a regret bound for the Thompson Sampling Horde of Bandits algorithm where the data is pooled using a known, prespecified, social graph. Vaswani et al. (2017) employ the conceptual framework of Agrawal and Goyal (2012) which uses the concept of *saturated* and *unsaturated* arms to bound the regret. They show that the regret for playing an arm from the unsaturated set (which includes the optimal arm) can be bounded by a factor of the standard deviation which decreases over time. Additionaly, they show that the probability of playing a saturated arm is small, so that an unsaturated arm will be played at each time with some constant probability. Vaswani et al. (2017) follow this argument, but adapt their proof to include the prior covariance of the social graph in the bound of the variance. Our proof follows along similar lines with the primary difference being how the prior covariance of all parameters is formulated. Specifically, the prior variance in Vaswani et al. (2017) is constructed by the Laplacian matrix of the social graph, whereas ours is constructed based on the Bayesian mixed effects model (4). As a result, while in Vaswani et al. (2017) the regret bound is stated in terms of properties of the social graph, our bound depends on properties of our mixed effects model (i.e., the covariance matrix of the random effects).

Recall that Σ_w is the prior covariance of the weight vector w_{pop}, Σ_u is the covariance of the random effect u_i and σ^2_ϵ is the variance of the error term. We assume that both w_{pop} and u_i have the same dimensions and that Σ_u is invertible. Additionally, for simplicity of presentation we assume that the largest eigenvalue in Σ_w is at most d and the largest eigenvalue of Σ_u is at most dN.

Recall that Theorem 1 bounds the regret of IntelligentPooling at time T by:

$$R(T) = \tilde{O}\left(dN\sqrt{T}\sqrt{\log\Big(\frac{(\mathrm{Tr}(\Sigma_w) + \mathrm{Tr}(\Sigma_u) + \mathrm{Tr}(\Sigma_u^{-1}))}{d} + \frac{T}{\sigma^2_\epsilon dN}\Big)\log\frac{1}{\delta}}\right)$$

with probability $1-\delta$.

Proof sketch of Theorem 1 We align the decision times from all users by the calendar time. Specifically, for a given time t, we retrieve the user index encountered at time t by $i(t)$ and retrieve this user's decision time index by $k(t)$. IntelligentPooling selects an action $A_{i(t),k(t)} \in \mathcal{A}$ for time $t \in [1, \ldots, T]$. We denote the selected action at time t by A_t.

In this setting, we combine each user specific variable into a global shared variable. Recall that a feature vector $\phi(A_{i,k}, S_{i,k})$ encodes contextual variables for the action and state of user i at their k^{th} decision time. For simplicity, we denote by A_t the action $A_{i(t),k(t)}$ at time t and denote the vector $\phi(A_{i(t),k(t)}, S_{i(t),k(t)})$ at time t by $\phi_{A_t,t}$. Additionally, we let $\phi_{a,t}$ refer to $\phi(a, S_{i(t),k(t)})$ for any $a \in \mathcal{A}$. We introduce a sparse vector $\boldsymbol{\varphi}_{A_t,t} \in \mathbb{R}^{dN}$, which contains $\phi_{A_t,t}$ vector among N d-dimensional vectors, the rest of which are zeros .

In proving the regret we consider the equivalent way of selecting the action. Instead of randomizing the action by the probability, here to select an action we assume the algorithm draws a sample $\tilde{w}_t = \tilde{w}_{i(t),k(t)}$ and then selects the action $A_t = A_{i(t),k(t)} = \underset{a\in\mathcal{A}}{\mathrm{argmax}}\, \phi^T_{a,t}\tilde{w}_t$ that maximizes the sampled reward. Analogously to $\phi_{a,t}$, we define $\hat{\mathbf{w}}_t$ and $\tilde{\mathbf{w}}_t$ as the sparse vectors which contain $\hat{w}_{i(t),k(t)}$ and $\tilde{w}_{i(t),k(t)}$ respectively as the $i(t)$-th vector among Nd-dimensional vector, the rest of which are zeros.

We concatenate the person-specific parameters w_i into $\mathbf{w} \in \mathbb{R}^{dN}$. Let the prior covariance of $\mathbf{w}$ be $\Sigma_0 = \mathbf{1}_{N\times N} \otimes \Sigma_w + \mathbf{I}_N \otimes \Sigma_u$. At time t, all contexts observed thus far, for all

users, can be combined into one matrix $\boldsymbol{\Phi}_t \in \mathbb{R}^{t\times dN}$ where a single row s corresponds to $\boldsymbol{\varphi}_{a_s,s}$, the sparse context vector associated with the action A_s taken for user $i(s)$ at their $k(s)$-th decision time. Let, $\Omega_t = \frac{1}{\sigma_\epsilon^2}\boldsymbol{\Phi}_t^\top \boldsymbol{\Phi}_t + \Sigma_0$. At each decision time t we draw a feature vector $\tilde{\mathbf{w}}_t \sim \mathcal{N}(\hat{\mathbf{w}}_t, v_t^2 \Omega_t^{-1})$.

Now, within this framework, we rewrite the instantaneous regret as $\Delta_t = \boldsymbol{\varphi}_{a_t^*,t}^\top \mathbf{w}_t - \boldsymbol{\varphi}_{A_t,t}^\top \mathbf{w}_t$. We prove that with high probability both $\boldsymbol{\varphi}_{a,t}^\top \hat{\mathbf{w}}_t$ and $\boldsymbol{\varphi}_{a,t}^\top \tilde{\mathbf{w}}_t$ are concentrated around their respective means. The standard deviation around the reward at decision time t for action a is thus $s_{a,t} = \sqrt{\boldsymbol{\varphi}_{a,t}^\top \Omega_{t-1}^{-1} \boldsymbol{\varphi}_{a,t}}$. We proceed as in Agrawal and Goyal (2012), Vaswani et al. (2017) by bounding three terms, the event $\mathcal{E}^{\theta_t}$, the event $\mathcal{E}^{\mathbf{w}_t}$ and $\sum_{t=1}^T s_{A_t,t}^2$

Definition 1 Let $\sigma_{u\min}^{-1}$ be the inverse of the smallest eigenvalue of Σ_u, $\sigma_{u\max}$ be the largest eigenvalue of Σ_u, $\sigma_{p\max}$ be the largest eigenvalue of Σ_w and let $\sigma_{\max} = \sigma_{u\max} + \sigma_{p\max}$. We assume that $\sigma_{u\max} \leq dN$ and $\sigma_{p\max} \leq d$.

Definition 2 For all a, define $\theta_{a,t} = \boldsymbol{\varphi}_{a,t}^\top \tilde{\mathbf{w}}_t$.

$$l_t = \sqrt{dN\log\Big(1 + \frac{\sigma_{\max}\sigma_{u\min}^{-1}}{\delta} + \frac{t\sigma_{u\min}^{-1}}{dN\delta}\Big)} + \sqrt{N\sigma_{p\max} + \sigma_{u\max}}$$

$$v_t = 2\sqrt{dN\log\Big(1 + \frac{\sigma_{\max}\sigma_{u\min}^{-1}}{\delta} + \frac{t\sigma_{u\min}^{-1}}{dN\delta}\Big)}$$

$$g_t = \min\{\sqrt{4dN\ln(t)}, \sqrt{4\ln(|\mathcal{A}|t)}\}v_t + l_T.$$

Definition 3

Definition 4 Define $\mathcal{E}^{\mathbf{w}_t}$ and $\mathcal{E}^{\theta_t}$ as the events that $\boldsymbol{\varphi}_t^\top \hat{\mathbf{w}}_t$ and $\theta_{A_t,t}$ are concentrated around their respective means. Recall that $|\mathcal{A}|$ is the total number of actions. Formally, define $\mathcal{E}^{\mathbf{w}_t}$ as the event that

$$\forall a : |\boldsymbol{\varphi}_{a,t}^\top \hat{\mathbf{w}}_t - \boldsymbol{\varphi}_{a,t}^\top \mathbf{w}| \leq l_t s_{a,t}.$$

Define $\mathcal{E}^{\theta_t}$ as the event that

$$\forall a : |\theta_{A_t,t} - \boldsymbol{\varphi}_{A_t,t}^\top \hat{\mathbf{w}}_t| \leq \min\{4dN\log(t), 4\log(|\mathcal{A}|t)\}v_t s_{a,t}.$$

Let $\zeta = \frac{1}{4e\sqrt{\pi}}$. Given that the events $\mathcal{E}^{\mathbf{w}_t}$ and $\mathcal{E}^{\theta_t}$ hold with high probability, we follow an argument similar to Lemma 4 of Agrawal and Goyal (2012) and obtain the following bound:

$$\mathcal{R}(T) \leq \frac{3g_T}{\zeta}\sum_{t=1}^T s_{A_t,t} + \frac{2g_T}{\zeta}\sum_{t=1}^T \frac{1}{t^2} + 6g_T\sqrt{|\mathcal{A}|T\log(2/\delta)}. \tag{15}$$

To bound the variance of the selected actions, $\sum_{t=1}^T s_{A_t,t}$, we follow an argument similar to Vaswani et al. (2017), and include the prior covariance terms of our model. We prove the following inequality:

$$\sum_{t=1}^T s_{A_t,t} \leq \sqrt{dNT}\sqrt{C\Big(\log\Big(\frac{(\mathrm{Tr}(\Sigma_w) + \mathrm{Tr}(\Sigma_u) + \mathrm{Tr}(\Sigma_u^{-1}))}{d} + \frac{T}{\sigma_\epsilon^2 dN}\Big)\Big)}, \tag{16}$$

where C is a constant equal to $\frac{\sigma_{umin}^{-1}}{\log(1+\frac{\sigma_{umin}^{-1}}{\sigma_\epsilon^2})}$. By combining Eqs. 15 and 16 we obtain the bound given in Theorem 1. □

Appendix 2: Supporting Lemmas

Definition 5 Recall that at time t we define as $\mathcal{D}_t$ as the history of all observed states, actions, and rewards up to time t. Define filtration $\mathcal{F}_{t-1}$ as the union of history until time $t-1$, and the contexts at time t, i.e., $\mathcal{F}_{t-1} = \{\mathcal{D}_{t-1}, \boldsymbol{\varphi}_{a,t}, a \in \mathcal{A}\}$. By definition, $\mathcal{F}_1 \subseteq \mathcal{F}_2 \cdots \subseteq \mathcal{F}_{t-1}$. The following quantities are also determined by the history $\mathcal{D}_{t-1}$ and the contexts, $\boldsymbol{\varphi}_{a,t}$ and are included in $\mathcal{F}_{t-1}$.

- $\hat{\mathbf{w}}_t, \Omega_{t-1}$
- $s_{a,t} \forall a$
- the identity of the optimal action a_t^*
- whether $\mathcal{E}_t^{\mathbf{w}}$ is true or not
- the distribution of $\mathcal{N}(\hat{\mathbf{w}}_t, v_t^2 \Omega_{t-1}^{-1})$

Note that the actual action A_t which is selected at decision point t is not included in $\mathcal{F}_{t-1}$. We now address the lemmas used in the proof which differ from Agrawal and Goyal (2012), Vaswani et al. (2017).

Lemma 1 *For* $\delta \in (0,1)$:

$$Pr(\mathcal{E}^{\mathbf{w}_t}) \geq 1 - \frac{\delta}{2}$$

Proof The true reward at time t, $R_t = \boldsymbol{\varphi}_{A_t,t}^\top \mathbf{w} + \epsilon_t$. Let, $\Omega_t \hat{\mathbf{w}}_t = \frac{\mathbf{b}_t}{\sigma_\epsilon^2}$. Define $\mathbf{S}_{t-1} = \sum_{l=1}^{t-1} \epsilon_l \boldsymbol{\varphi}_{a_l,l}$.

$$\mathbf{S}_{t-1} = \sum_{l=1}^{t-1} (R_l - \boldsymbol{\varphi}_{a_l,l}^\top \mathbf{w}) \boldsymbol{\varphi}_{a_l,l} = \sum_{l=1}^{t-1} (R_l \boldsymbol{\varphi}_{a_l,l} - \boldsymbol{\varphi}_{a_l,l} \boldsymbol{\varphi}_{a_l,l}^\top \mathbf{w})$$

$$\mathbf{S}_{t-1} = b_{t-1} - \sum_{l=1}^{t-1} (\boldsymbol{\varphi}_{a_l,l} \boldsymbol{\varphi}_{a_l,l}^\top \mathbf{w}) = b_{t-1} - \sigma_\epsilon^2 (\Omega_{t-1} \hat{\mathbf{w}}_t - \Omega_{t-1} \mathbf{w} + \Sigma_0 \mathbf{w})$$

$$\hat{\mathbf{w}}_t - \mathbf{w} = \Omega_{t-1}^{-1} \big(\frac{\mathbf{S}_{t-1}}{\sigma_\epsilon^2} - \Sigma_0 \mathbf{w} \big).$$

The following holds for all a:

$$\begin{aligned} |\boldsymbol{\varphi}_{a,t}^\top \hat{\mathbf{w}}_t - \boldsymbol{\varphi}_{a,t}^\top \mathbf{w}| &= |\boldsymbol{\varphi}_{a,t}^\top (\hat{\mathbf{w}}_t - \mathbf{w})| \\ &\leq \big| \boldsymbol{\varphi}_{a,t} \Omega_{t-1}^{-1} \big(\frac{\mathbf{S}_{t-1}}{\sigma_\epsilon^2} - \Sigma_0 \mathbf{w} \big) \big| \\ &\leq \| \boldsymbol{\varphi}_{a,t} \|_{\Omega_{t-1}^{-1}} \Big(\Big\| \frac{\mathbf{S}_{t-1}}{\sigma_\epsilon^2} - \Sigma_0 \mathbf{w} \Big\|_{\Omega_{t-1}^{-1}} \Big). \end{aligned}$$

By the triangle inequality,

$$|\boldsymbol{\varphi}_{a,t}^{\top}\hat{\mathbf{w}}_t - \boldsymbol{\varphi}_{a,t}^{\top}\mathbf{w}| \leq \left(\left\|\frac{\mathbf{S}_{t-1}}{\sigma_\epsilon^2}\right\|_{\Omega_{t-1}^{-1}} + \|\Sigma_0\mathbf{w}\|_{\Omega_{t-1}^{-1}}\right) \tag{17}$$

We now bound the term $\|\Sigma_0\mathbf{w}\|_{\Omega_{t-1}^{-1}}$. Recall that the prior covariance of $\mathbf{w}$, $\Sigma_0 = \mathbf{1}_{N\times N} \otimes \Sigma_w + \mathbf{I}_N \otimes \Sigma_u$.

$$\begin{aligned}
\nu_{\max}(\Sigma_0) &= \nu_{\max}(\mathbf{1}_{N\times N} \otimes \Sigma_w + \mathbf{I}_N \otimes \Sigma_u) \\
&= \nu_{\max}(\mathbf{1}_{N\times N}) \cdot \nu_{\max}(\Sigma_w) + \nu_{\max}(\mathbf{I}_N) \cdot \nu_{\max}(\Sigma_u) \\
&= N\nu_{\max}(\Sigma_w) + \nu_{\max}(\Sigma_u) \\
&= N\sigma_{p\max} + \sigma_{u\max} \\
\|\Sigma_0\mathbf{w}\|_{\Omega_{t-1}^{-1}} &\leq \|\Sigma_0\mathbf{w}\|_{\Sigma_0^{-1}} = \sqrt{\mathbf{w}\Sigma_0^{\top}\Sigma_0^{-1}\Sigma_0\mathbf{w}} \qquad = \sqrt{\mathbf{w}^{\top}\Sigma_0\mathbf{w}} \\
&\leq \sqrt{\nu_{\max}(\Sigma_0)\|\mathbf{w}\|_2} \\
&\leq \sqrt{\nu_{\max}(\Sigma_0)} \\
&\leq \sqrt{N\sigma_{p\max} + \sigma_{u\max}}
\end{aligned}$$

For bounding $\|\boldsymbol{\varphi}_{a,t}\|_{\Omega_{t-1}^{-1}}$, note that

$$\|\boldsymbol{\varphi}_{a,t}\|_{\Omega_{t-1}^{-1}} = \sqrt{\boldsymbol{\varphi}_{a,t}^{\top}\Omega_{t-1}^{-1}\boldsymbol{\varphi}_{a,t}} = s_{a,t}.$$

We can thus write Eq. 17

$$|\boldsymbol{\varphi}_{a,t}^{\top}\hat{\mathbf{w}}_t - \boldsymbol{\varphi}_{a,t}^{\top}\mathbf{w}| \leq s_{a,t}\left(\frac{1}{\sigma_\epsilon}\left\|\mathbf{S}_{t-1}\right\|_{\Omega_{t-1}^{-1}} + \sqrt{n\sigma_{p\max} + \sigma_{u\max}}\right) \tag{18}$$

We now bound $\left\|\mathbf{S}_{t-1}\right\|_{\Omega_{t-1}^{-1}}$.

Theorem 2 *For any $d > 0, t \geq 1$, with probability at least $1-\delta$,*

$$\begin{aligned}
\left\|\mathbf{S}_{t-1}\right\|_{\Omega_{t-1}^{-1}}^2 &\leq 2\sigma_\epsilon^2\log\left(\frac{\det\Omega_t^{\frac{1}{2}}\det\Sigma_0^{\frac{-1}{2}}}{\delta}\right) \\
&\leq 2\sigma_\epsilon^2\Big(\log(\det\Omega_t^{\frac{1}{2}}) + \log(\det\Sigma_0^{\frac{-1}{2}}) - \log(\delta)\Big) \\
&\leq \sigma_\epsilon^2\Big(\log(\det\Omega_t) + \log(\det\Sigma_0{}^{-1}) - 2\log(\delta)\Big).
\end{aligned}$$

For any $n\times n$ matrix A, $\det(A) \leq \left(\frac{\mathrm{Tr}(A)}{n}\right)^n$. This implies, $\log(\det(A)) \leq n\log\left(\frac{\mathrm{Tr}(A)}{n}\right)$. Applying this inequality for both Ω_t and Σ_0^{-1}, we obtain:

$$\left\|\mathbf{S}_{t-1}\right\|_{\Omega_{t-1}^{-1}} \leq dN\sigma_\epsilon^2\left(\log\left(\frac{\mathrm{Tr}(\Omega_t)}{dN}\right) + \log\left(\frac{\mathrm{Tr}(\Sigma_0^{-1})}{dN}\right) - \frac{2}{dN}\log(\delta)\right) \tag{19}$$

Next, we use the fact that

$$\begin{aligned}\Omega_t = \Sigma_0 + \Sigma_{l=1}^{t} \boldsymbol{\varphi}_{a_l,l} \boldsymbol{\varphi}_{a_l,l}^{\top} \Rightarrow \mathrm{Tr}(\Omega_t) \leq \mathrm{Tr}(\Sigma_0) + t \\ \mathrm{Tr}(\Sigma_0) = \mathrm{Tr}(\mathbf{1}_{N\times N} \otimes \Sigma_w + \mathbf{I}_N \otimes \Sigma_u) \\ = \mathrm{Tr}(\mathbf{1}_{N\times N}) \cdot \mathrm{Tr}(\Sigma_w) + \mathrm{Tr}(\mathbf{I}_N) \cdot \mathrm{Tr}(\Sigma_u) \\ = N\mathrm{Tr}(\Sigma_w) + N\mathrm{Tr}(\Sigma_u) = N(\mathrm{Tr}(\Sigma_w) + \mathrm{Tr}(\Sigma_u))\end{aligned}$$

We now return to Eq. 19

$$\begin{aligned}\left\|\mathbf{S}_{t-1}\right\|_{\Omega_{t-1}^{-1}}^2 &\leq dN\sigma_\epsilon^2\left(\log\left(\frac{\mathrm{Tr}(\Sigma_0)+t}{dN}\right) + \log\left(\frac{\mathrm{Tr}(\Sigma_0^{-1})}{dN}\right) - \frac{2}{dN}\log(\delta)\right) \\ &\leq dN\sigma_\epsilon^2\left(\log\left(\frac{\mathrm{Tr}(\Sigma_0)\mathrm{Tr}(\Sigma_0^{-1}) + t\mathrm{Tr}(\Sigma_0^{-1})}{d^2N^2}\right) - \log(\delta^{\frac{2}{dN}})\right) \\ &= dN\sigma_\epsilon^2\left(\log\left(\frac{\mathrm{Tr}(\Sigma_0)\mathrm{Tr}(\Sigma_0^{-1}) + t\mathrm{Tr}(\Sigma_0^{-1})}{d^2N^2\delta}\right)\right) \\ &\leq dN\sigma_\epsilon^2\left(\log\left(\frac{d^2N^2\sigma_{\max}\sigma_{u\min}^{-1} + tdN\sigma_{u\min}^{-1}}{d^2N^2\delta}\right)\right) \\ &= dN\sigma_\epsilon^2\left(\log\left(\frac{\sigma_{\max}\sigma_{u\min}^{-1}}{\delta} + \frac{t\sigma_{u\min}^{-1}}{dN\delta}\right)\right) \\ \left\|\mathbf{S}_{t-1}\right\|_{\Omega_{t-1}^{-1}} &\leq \sigma_\epsilon\sqrt{dN\log\left(\frac{\sigma_{\max}\sigma_{u\min}^{-1}}{\delta} + \frac{t\sigma_{u\min}^{-1}}{dN\delta}\right)} \\ \left\|\mathbf{S}_{t-1}\right\|_{\Omega_{t-1}^{-1}} &\leq \sigma_\epsilon\sqrt{dN\log\left(1 + \frac{\sigma_{\max}\sigma_{u\min}^{-1}}{\delta} + \frac{t\sigma_{u\min}^{-1}}{dN\delta}\right)} \\ |\boldsymbol{\varphi}_{a,t}^{\top}\hat{\mathbf{w}}_t - \boldsymbol{\varphi}_{a,t}^{\top}\mathbf{w}| &\leq s_{a,t}\sqrt{dN\log\left(1 + \frac{\sigma_{\max}\sigma_{u\min}^{-1}}{\delta} + \frac{t\sigma_{u\min}^{-1}}{dN\delta}\right)} + \sqrt{N\sigma_{p\max} + \sigma_{u\max}} \\ &\leq s_{a,t}l_t\end{aligned}$$

□

Lemma 2 *With probability* $1 - \frac{\delta}{2}$,

$$\sum_{t=1}^{T}\mathrm{regret}(t) \leq \sum_{t=1}^{T}\frac{3g_t}{\zeta}s_t + \sum_{t=1}^{T}\frac{2g_t}{\zeta t^2}s_t + \sqrt{2\sum_{t=1}^{T}\frac{36g_t^2}{\zeta^2}\ln(\frac{2}{\delta})} \tag{20}$$

Proof Let Z_l and Y_t be defined as follows:

$$Z_l = \mathrm{regret}(l) - \frac{3g_l}{\zeta}s_l - \frac{2g_l}{\zeta l^2}s_l$$

$$Y_l = \sum_{l=1}^{t} Z_l$$

Hence, Y_t is a super-martingale process:

$$\mathbb{E}[Y_t - Y_{t-1}|\mathcal{F}_{t-1}] = \mathbb{E}[Z_t] = \mathbb{E}[\text{regret}(t)||\mathcal{F}_{t-1}] - \frac{3g_l}{\zeta}s_l - \frac{2g_l}{\zeta l^2}s_l$$

$$\mathbb{E}[\text{regret}(t)|\mathcal{F}_{t-1}] \leq \mathbb{E}[\Delta_t|\mathcal{F}_{t-1}] \leq \frac{3g_l}{\zeta}s_l + \frac{2g_l}{\zeta l^2}s_l$$

$$\mathbb{E}[Y_t - Y_{t-1}|\mathcal{F}_{t-1}] \leq 0$$

We now apply Azuma-Hoeffding inequality. We define $Y_0 = 0$. Note that $|Y_t - Y_{t-1}| = |Z_l|$ is bounded by $1 + 3g_l - 2g_l$. Hence, $c = 6g_t$. Setting $a = \sqrt{2\ln(\frac{2}{\delta})\sum_{t=1}^{T} c_t^2}$ in the above inequality, we obtain that with probability $1 - \frac{\delta}{2}$,

$$Y_t \leq \sqrt{2\ln(\frac{2}{\delta})\sum_{t=1}^{T} 36g_t^2} \tag{21}$$

$$\sum_{t=1}^{T}\left(\text{regret}(t) - \frac{3g_t}{\zeta}s_t - \frac{2g_t}{\zeta t^2}s_t\right) \leq \sqrt{2\ln(\frac{2}{\delta})\sum_{t=1}^{T} 36g_t^2} \tag{22}$$

$$\sum_{t=1}^{T}\left(\text{regret}(t)\right) \leq \sum_{t=1}^{T}\frac{3g_t}{\zeta}s_t + \sum_{t=1}^{T}\frac{2g_t}{\zeta t^2}s_t + \sqrt{2\ln(\frac{2}{\delta})\sum_{t=1}^{T} 36g_t^2} \tag{23}$$

□

Lemma 3 (Azuma-Hoeffding). *If a super-martingale Y_t (with $t \geq 0$) and its the corresponding filtration $\mathcal{F}_{t-1}$, satisfies $|Y_t - Y_{t-1}| \leq ct$ for some constant c for all $t = 1, \ldots, T$ then for any $x \geq 0$:*

$$Pr(Y_t - Y_0 \geq x) \leq exp\left(\frac{-x^2}{2\sum_{t=1}^{T} c_t^2}\right) \tag{24}$$

Lemma 4 $\sum_{t=1}^{T} s_{A_t,t} \leq \sqrt{dNT}\sqrt{C\left(\log\left(\frac{(\text{Tr}(\Sigma_w)+\text{Tr}(\Sigma_u)+\text{Tr}(\Sigma_u^{-1}))}{d} + \frac{T}{\sigma_\epsilon^2 dN}\right)\right)}$

For simplicity, we let $s_{A_t,t} = s_t$ below.

$$
\begin{aligned}
\det|\mathbf{1}_{N\times N}\otimes\Sigma_w|+\det|\mathbf{I}_N\otimes\Sigma_u|) &= \det|\mathbf{1}_{N\times N}|^d\det|\Sigma_w|^N+\det|\mathbf{I}_N|^d\det|\Sigma_u|^N\\
&= \det|\Sigma_u|^N\\
\log(\det|\Omega_t|) &\geq \log(\det|\Sigma_0|)+\sum_{t=1}^{T}\log(1+\frac{s_t^2}{\sigma_\epsilon^2})\\
&\geq \log(\det|\mathbf{1}_{N\times N}\otimes\Sigma_w|+\det|\mathbf{I}_N\otimes\Sigma_u|)+\sum_{t=1}^{T}\log(1+\frac{s_t^2}{\sigma_\epsilon^2})\\
&= n\log(\det|\Sigma_u|)+\sum_{t=1}^{T}\log(1+\frac{s_t^2}{\sigma_\epsilon^2})\\
\mathrm{Tr}(\Omega_t) &\leq \mathrm{Tr}(\Sigma_0)+\frac{T}{\sigma_\epsilon^2}\\
&= \mathrm{Tr}(\mathbf{1}_{N\times N}\otimes\Sigma_w)+\mathrm{Tr}(\mathbf{I}_N\otimes\Sigma_u)+\frac{T}{\sigma_\epsilon^2}\\
&= \mathrm{Tr}(\mathbf{1}_{N\times N})\mathrm{Tr}(\Sigma_w)+\mathrm{Tr}(\mathbf{I}_N)\mathrm{Tr}(\Sigma_u)+\frac{T}{\sigma_\epsilon^2}\\
&= N\mathrm{Tr}(\Sigma_w)+N\mathrm{Tr}(\Sigma_u)+\frac{T}{\sigma_\epsilon^2}
\end{aligned}
$$

Using the determinant-trace inequality, we have the following relation:

$$\Big(\frac{1}{dN}\mathrm{Tr}(\Omega_t)\Big)^{dN} \geq \det|\Omega_t|$$

$$dN\log(\frac{1}{dN}\mathrm{Tr}(\Omega_t)) \geq \log(\det|\Omega_t|)$$

$$dN\log(\frac{1}{dN}\mathrm{Tr}(\Omega_t)) \geq \log(\det|\Omega_t|)$$

$$dN\log(\frac{1}{dN}(\mathrm{Tr}(\Sigma_0)+\frac{T}{\sigma_\epsilon^2})) \geq \log(\det|\Omega_t|) \geq N\log(\det|\Sigma_u|)+\sum_{t=1}^{T}\log(1+\frac{s_t^2}{\sigma_\epsilon^2})$$

$$dN\log(\frac{1}{dN}(\mathrm{Tr}(\Sigma_0)+\frac{T}{\sigma_\epsilon^2})) \geq N\log(\det|\Sigma_u|)+\sum_{t=1}^{T}\log(1+\frac{s_t^2}{\sigma_\epsilon^2})$$

$$dN\log(\frac{1}{dN}(\mathrm{Tr}(\Sigma_0)+\frac{T}{\sigma_\epsilon^2}))-N\log(\det|\Sigma_u|) \geq \sum_{t=1}^{T}\log(1+\frac{s_t^2}{\sigma_\epsilon^2})$$

$$dN\log(\frac{1}{dN}(\mathrm{Tr}(\Sigma_0)+\frac{T}{\sigma_\epsilon^2}))+N\log(\det|\Sigma_u^{-1})| \geq \sum_{t=1}^{T}\log(1+\frac{s_t^2}{\sigma_\epsilon^2})$$

$$dN\log(\frac{1}{dN}(\mathrm{Tr}(\Sigma_0)+\frac{T}{\sigma_\epsilon^2}))+dN\log(\frac{1}{d}\mathrm{Tr}(\Sigma_u^{-1})) \geq \sum_{t=1}^{T}\log(1+\frac{s_t^2}{\sigma_\epsilon^2})$$

$$dN\Big(\log(\frac{1}{dN}(\mathrm{Tr}(\Sigma_0)+\frac{T}{\sigma_\epsilon^2}))+\log(\frac{1}{d}\mathrm{Tr}(\Sigma_u^{-1}))\Big) \geq \sum_{t=1}^{T}\log(1+\frac{s_t^2}{\sigma_\epsilon^2})$$

$$dN\Big(\log((\frac{\mathrm{Tr}(\Sigma_0)\sigma_\epsilon^2+T}{\sigma_\epsilon^2 dN}))+\log(\frac{1}{d}\mathrm{Tr}(\Sigma_u^{-1}))\Big) \geq \sum_{t=1}^{T}\log(1+\frac{s_t^2}{\sigma_\epsilon^2})$$

$$dN\Big(\log(\frac{\mathrm{Tr}(\Sigma_0)\sigma_\epsilon^2+T+N\mathrm{Tr}(\Sigma_u^{-1})\sigma_\epsilon^2}{\sigma_\epsilon^2 dN})\Big) \geq \sum_{t=1}^{T}\log(1+\frac{s_t^2}{\sigma_\epsilon^2})$$

$$dN\Big(\log(\frac{(n\mathrm{Tr}(\Sigma_w)+N\mathrm{Tr}(\Sigma_u))\sigma_\epsilon^2+T+N\mathrm{Tr}(\Sigma_u^{-1})\sigma_\epsilon^2}{\sigma_\epsilon^2 dN})\Big) \geq \sum_{t=1}^{T}\log(1+\frac{s_t^2}{\sigma_\epsilon^2})$$

$$dN\Big(\log(\frac{(\mathrm{Tr}(\Sigma_w)+\mathrm{Tr}(\Sigma_u)+\mathrm{Tr}(\Sigma_u^{-1}))}{d}+\frac{T}{\sigma_\epsilon^2 dN})\Big) \geq \sum_{t=1}^{T}\log(1+\frac{s_t^2}{\sigma_\epsilon^2})$$

Let, $s_t^2 \leq \sigma_{u\min}^{-1}$. For all $y \in [0, \sigma_{u\min}^{-1}]\log(1+\frac{y}{\sigma_\epsilon^2}) \geq \frac{1}{\sigma_{u\min}^{-1}}\log(1+\frac{\sigma_{u\min}^{-1}}{\sigma_\epsilon^2})y$
(See argument in Vaswani et al. (2017)).

$$\log(1+\frac{s_t^2}{\sigma^2}) \geq \frac{1}{\sigma_{u\min}^{-1}}\log(1+\frac{\sigma_{u\min}^{-1}}{\sigma_\epsilon^2})s_t^2)$$

$$\frac{1}{\sigma_{u\min}\log(1+\frac{1}{\sigma_{u\min}\sigma_\epsilon^2})}\log(1+\frac{s_t^2}{\sigma_\epsilon^2}) \leq s_t^2$$

$$\sum_{t=1}^{T}s_t^2 \leq C\sum_{t=1}^{T}\log(1+\frac{s_t^2}{\sigma_\epsilon^2})$$

Where, $C = \sigma_{u\min}\log(1 + \frac{1}{\sigma_{u\min}\sigma_\epsilon^2})$

By Cauchy Schwartz

$$\sum_{t=1}^{T} s_t \leq \sqrt{T}\sqrt{\sum_{t=1}^{T} s_t^2}$$

$$\sum_{t=1}^{T} s_t \leq \sqrt{T}\sqrt{C\sum_{t=1}^{T}\log(1 + \frac{s_t^2}{\sigma_\epsilon^2})}$$

$$\sum_{t=1}^{T} s_t \leq \sqrt{T}\sqrt{CdN\Big(\log\Big(\frac{(\mathrm{Tr}(\Sigma_w) + \mathrm{Tr}(\Sigma_u) + \mathrm{Tr}(\Sigma_u^{-1}))}{d} + \frac{T}{\sigma_\epsilon^2 dN}\Big)\Big)}$$

$$\sum_{t=1}^{T} s_t \leq \sqrt{dNT}\sqrt{C\Big(\log\Big(\frac{(\mathrm{Tr}(\Sigma_w) + \mathrm{Tr}(\Sigma_u) + \mathrm{Tr}(\Sigma_u^{-1}))}{d} + \frac{T}{\sigma_\epsilon^2 dN}\Big)\Big)}$$

□

Appendix 3: Simulation

We include additional information about the simulation environment. We first explain general information about the simulation environment. We then provide the procedures for generating state variables (features) in the simulation. Finally, we discuss how we used HeartStepsV1 to arrive at the feature representations used in the simulation.

Simulation dynamics Within the simulation states are updated every thirty minutes. Each thirty minutes is associated with a date-time, thus we can acquire the month from the current time which is useful in updating the temperature. The decision times are set roughly two hours apart from 9:00 to 19:00.

Availability In the real-study users are not always available to receive treatment for a suite of reasons. For example, they may be driving a vehicle or they might have recently received treatment. Thus, at each decision time we update the context feature $Available_i \sim Bernoulli(.8)$. for the i^{th} user where $Available_i$ is drawn from a Bernoulli. This condition reduces the distance between the settings in the environment and those in a real-world study. At each decision time interventions are only sent to users who are available; i.e. user i cannot receive an intervention when $Available_i = 0$.

Recruitment We follow the recruitment rate observed in HeartStepsV1. For example, if 20% of the total number of participants were recruited in the third week of HeartStepsV1 we recruit 20% of the total number of participants who will be recruited in the third week of the simulation. To explore the effect of running the study for varying lengths we scale the recruitment rates. For example, if the true study ran for 8 weeks, and we want to run a simulation for three weeks, we proportionally scale the recruitment in each of the three weeks so that the relative recruitment in each week remains the same. In these experiments we would like to recruit the entire population within 6 weeks. Thus about 10% of participants are recruited each week, except for the second week of the study where about 30% of all participants are recruited. This reflects the recruitment rates seen in the study, which were more of less consistent throughout besides one increase in the second week.

We generate states from historical data. Given relevant context we search historical data for states which match this given context. This subset of matching states can be used to generate new states. We discuss this in more detail in Sect. C.1. Then, we describe in more detail how we generate temperature, location and step counts.

Querying history

Algorithm 2 is used to obtain relevant historical data in order to form a probability distribution over some target feature value. For example, if we would like a probability distribution over discretized temperature IDs under a given context, we would search over the historical data for all temperature IDs present under this context. This set of context-specific temperature IDs can then be used to form a distribution to simulate a new ID. This process of querying historical data is used throughout the simulation and is outlined in Algorithm 2. For example, it is used in generating new step counts, new locations and new temperatures.

Algorithm 2 QUERYHISTORY

INPUT = historical data $[\mathbf{x_i}; i = [1, N]]$, conditioning state $\mathbf{x}^*$, target data variable $y = f(x)$,
$\mathcal{S} = \{\}$
for $i = 1$ to N **do**
  if $\mathbf{x_i} == \mathbf{x}^*$ **then**
    Add $f(x_i)$ to $\mathcal{S}$
  end if
end for
OUTPUT $=\mathcal{S}$

As the simulation environment simulates draws stochastically from a variety of probability distributions, it is possible it draws a state which was not present in the historical dataset. In this case there is a process for finding a matching state. Similarly we might have a state in the historical dataset with insufficient samples to form an informative (not overly-noisy) distribution. In this case we also find a surrogate state with which to generate future step counts. The idea of the process is to find the closest state to the current state, such that this close state has sufficient data to generate a good distribution. Again, given a state, we want to be able to generate a step count from a distribution with sufficient data to inform its parameters. The pseudocode for how we do so is shown in Algorithm 3

This algorithm takes as input a target state, s^*. We also have a dictionary(hasmap) formed from the historical dataset. The keys to this dictionary are the states which existed in the dataset. A value is an array of step counts for this state.

Algorithm 3 FindMatch

```
1: INPUT = current state s* ∈ ℝ^d, dictionary of existing states to step counts 𝔻 = {s : [c_1, ..., c_N]}
2: match←None
3: if s* ∈ 𝔻 and len(𝔻[s*]) > 30 then
4:    match← s*
5: else
6:    new_size = d-1
7:    while match is None do
8:       #find state of size new size with most data points in historical dataset
9:       form new states of size new_size
10:      rank states s by len(𝔻[s])
11:      choose state with greatest len
12:      temp ← max_s len(𝔻[s])
13:      if 𝔻[temp] > 30 then
14:         match← temp
15:      end if
16:      new_size = new_size − 1
17:   end while
18: end if
```

This procedure gives the closest state with the most data points to our current state.

To be more explicit about lines 8-11. A state is a vector of some length, for example [1, 0, 1]. When we consider all subsets of size 2, we are considering the subsets [1, 0],[1, 1], and [0, 1]. For each of these we can look in the historical data set and find all points where this state was true. Thus for each subset we'll get a new list of points, $[1,0] = [c_1, \ldots, c_{N1}]$ $[1,1] = [c_1, \ldots, c_{N2}]$, $[0,1] = [c_1, \ldots, c_{N3}]$. We now look at $N1$, $N2$, $N3$ and choose the state with the highest value. For example, if the lists were: $[1,0] = [c_1, \ldots, c_{100}]$ $[1,1] = [c_1, \ldots, c_2]$, $[0,1] = [c_1, \ldots, c_{300}]$, we would choose $s = [0,1]$. Now if we encounter the state [1, 0, 1] and there is insufficient data to form a distribution from this state, we will instead form it from the values found under the state [0, 1], $[c_1, \ldots, c_{300}]$.

Generating temperature

We mimic a trial where everyone resides in the same general area, such as a city. In this setting everyone experiences the same global temperature. We describe how to obtain temperature at any point in time in Algorithm 4. The temperature is updated exactly five times a day.

In the following algorithms t, refers to a timestamp, $\mathcal{D}$ refers to a historical dataset, K_t refers to a set of temperature IDs, and w_{t-1} refers to the temperature at the previous time stamp. Here, $\mathcal{D}$ = HeartStepsV1 and $K_t = \{\text{hot}, \text{cold}\}$. The contextual features which influence temperature are time of day, day of the week and the month *tod*, *dow* and *month* respectively. Furthermore, at all times besides the first moment in the trial, the next temperature depends on the current temperature w_{t-1}.

Algorithm 4 GetTemperature

INPUT $= t, \mathcal{D}, K_t, w_{t-1},$
$tod \leftarrow tod(t)$
$dow \leftarrow dow(t)$
$month \leftarrow month(t)$
if w_{t-1} is Null **then**
  $q \leftarrow [tod, dow, month]$
else
  $q \leftarrow [tod, dow, month, w_{t-1}]$
end if
$\mathbf{p} \leftarrow [0]_{K_l}$
$\mathcal{T} \leftarrow$ QueryHistory$(\mathcal{D}, q, w)$
for $k \in K_t$ **do**
  $p_k = \frac{1}{|\mathcal{T}|} \sum_{i=0}^{|\mathcal{T}|} \mathbb{1}_{l_i == k}$
end for
$w_t \sim Categorical([p_{cold}, p_{hot}])$
OUTPUT w_t

Generating location

In the following algorithms t, refers to a timestamp, g_u refers to the group id of user i,$\mathcal{D}$ refers to a historical dataset, K_t refers to a set of location IDs, and l_{t-1} refers to the location at the previous time stamp. Here, $\mathcal{D}$ = HeartStepsV1 and $K_t = \{$other, home or work$\}$.

As in generating temperature, the contextual features which influence location are time of day, day of the week and the month *tod*, *dow* and *month* respectively. Generating location is different from generating temperature in that each user moves from location to location independently. Whereas we model users to share one common temperature, they move from one location to another independently of other users. Thus we also include group id in determining the next location for a given user.

1. User is at a decision time
   (a) User is available
   (b) User is not available
2. User is not at a decision time

Generating step-counts

A new step-count is generated for each User active in the study, every thirty-minutes according to one of the following scenarios:

Algorithm 5 GetLocation

INPUT $= t, g_u, \mathcal{D}, K_l$
$tod \leftarrow tod(t)$
$dow \leftarrow dow(t)$
Find t_0 in $\mathcal{D}$
if l_{t-1} is Null **then**
  $q \leftarrow [tod, dow, g_u]$
else
  $q \leftarrow [tod, dow, g_u, l_{t-1}]$
end if
$\mathcal{L} \leftarrow$ QueryHistory$(\mathcal{D}, q, l)$
$\mathbf{p} \leftarrow [0]_{K_l}$
for $k \in K_l$ **do**
  $p_k = \frac{1}{|\mathcal{L}|} \sum_{i=0}^{|\mathcal{L}|} \mathbb{1}_{l_i == k}$
end for
$l_t \sim Categorical([p_{\text{other}}, p_{\text{home or work}}])$
OUTPUT l_t

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). Recall, that if a user is available the final step count is generated according to Eq. 25. This equation requires sufficient statistics from HeartStepsV1. The procedure for obtaining these statistics is shown explicitly in Algorithm 6.

$$R_{i,k} = \mathbf{N}(\mu_{h(S_{i,k})}, \sigma^2_{h(S_{i,k})}) + A_{i,k}(f(S_{i,k})^T \beta_i + Z_i). \tag{25}$$

Here, $t, g^u, w_t, l_u, \mathcal{D}$ refer to the current time in the trial, the group id of the i^{th} user, the temperature at time t, the location of the i^{th} user, and a historical dataset, respectively. To find sufficient statistics of step counts, we also employ the time of day and day of the week, *tod* and *dow* respectively. Finally, $yst(t, u)$ describes the previous step count as high or low.

Algorithm 6 StepStatistics

INPUT $= t, g^u, w_t, u, \mathcal{D}$
#Compute variables included in conditioning context
$tod \leftarrow tod(t)$
$dow \leftarrow dow(t)$
$y \leftarrow yst(t, u)$
$q \leftarrow [g^u, w_t, tod, dow, y, l_{t,u}, a]$
#Obtain step counts from $\mathcal{D}$ conditioned on q
$\mathcal{S} \leftarrow$ QueryHistory$(\mathcal{D}, q, c)$
$\hat{\mu}_{\mathcal{S}} \leftarrow \frac{1}{|\mathcal{S}|} \sum_{i=0}^{|\mathcal{S}|} s_i$
$\hat{\sigma}^2_{\mathcal{S}} \leftarrow \frac{1}{|\mathcal{S}|} \sum_{i=0}^{|\mathcal{S}|} (s_i - \hat{\mu}_{\mathcal{S}})^2$
OUTPUT $\hat{\mu}_{\mathcal{S}}, \hat{\sigma}^2_{\mathcal{S}}$

Settings for Heterogeneity

Appendix 4: Feature construction

We provide more details on the processes used for feature construction. As stated in the paper we rely heavily on the dataset HeartStepsV1 to make all feature construction decisions. The one exception is in the design of the location feature, for which we had domain knowledge to rely on (more detail below)

Baseline activity

Each user is assigned to one of two groups: a low-activity group or a high-activity group. These groups are found from the historical data. We perform hierarchical clustering using the method *hcluster* in scikit-learn Pedregosa (2011). We used a euclidean distance metric to cluster the data and found that two groups naturally arose. These groups were consistent with the population of HeartStepsV1, which consisted of participants who were generally either office administrators or students.

State features

We now briefly outline the decisions for the remaining features: time of day, day of the week, and temperature. For each feature we explored various categorical representations. For each, the question was how many categories to use to represent the data. For each feature we followed the same procedure.

1. We chose a number of categories (k) to threshold the data into
2. We partitioned the data into k categories
3. We clustered the step counts according to these k categories
4. We computed the Calinski-Harabasz score of this clustering
5. We chose the final k to be that which provided the highest score

For example, consider the task of representing temperature. Let l be a temperature, x be a step count and x_{l_b} be a thirty-minute step count occurring when the temperature l was assigned to bucket b. Given a historical dataset, we have a vector $\mathbf{x}$ where each entry $x_{i,t}$ refers to the thirty-minute step count of user i at time t.

- Let p be a number of buckets. We create p buckets by finding quantiles of l. For example, if p=2, we find the 50^{th} quantile of l. A bucket is defined by a tuple of thresholds (th_1, th_2), such that for a data point d to belong to bucket i, d must be in the range of the tuple $(th_1 \leq d < th_2)$.
- For each temperature, we determine the bucket label which best describes this temperature. That is the label y of l, is the bucket for which $th_1^y \leq \bar{s}^l < th_2^y$.
- We now create a vector of labels y, of the same length as $\mathbf{x}$. Each $y_{i,t}^l$ is the bucket assigned to $l_{i,t}$. For example, if the temperature for user i at time t falls into the lowest bucket, 0 would be the label assigned to $l_{i,t}$. This induces a clustering of step-counts where the label is a temperature bucket.
- We determine the Calabrinski-Harabasz score of this clustering.

We test this procedure from p equal to 1, through 4.

Table 6 Settings for Z in three cases of homogeneous, bimodal and smoothly varying populations

Homogeneous	Bi-modal	Smooth
$Z^i = 0\ \beta_i^l{=}0$	$Z_i, \beta_i^l = \begin{cases} 0.1, 0.l & \text{if } i \in \text{group one} \\ -0.3, -0.l & \text{if } i \in \text{group two} \end{cases}$	$Z_i \sim \mathcal{N}(0, 0.35)\ \beta_i^l \sim \mathcal{N}(0, 0.1)$

For example, consider determining a representation for time of day. We choose a partition to be morning, afternoon, evening. For each thirty-minute step count, if it occurred in the morning we assign it to the morning cluster, if it occurred in the afternoon we assign it to the afternoon cluster, etc. Now we have three clusters of step counts and we can compute the C score of this clustering. We repeat the process for different partitions of the day.

Time of day To discover the representation for time of day which best explained the observed step counts, we considered all sequential partitions from length 2-8. We found that early-day, late-day, and night best explained the data.

Day of the week To discover the representation for day of the week which best explained the observed step counts, we considered two partitions: every day, or weekday/weekend. We found weekday/weekend to be a better fit to the data.

Temperature Here we choose different percentiles to partition the data. We consider between 2 and 5 partitions (percentiles at 50, to 20,40,60,80). Here we found two partitions to best fit the step counts. We also tried more complicated representations of weather combined with temperature, however for the purpose of this paper we found a simple representation to best allow us to explore the relevant questions in this problem setting.

Location In representing location we relied on domain knowledge. We found that participants tend to be more responsive when they are either at home or work, than in other places. Thus, we decided to represent location as belonging to one of two categories: home/work or other.

Appendix 5: Feasibility study

In the clinical trial we describe users' states with the features described in Table 4. The two features which differ from the simulation environment are engagement and exposure to treatment. We clarify these features below (Table 6).

Engagement The engagement variable measures the extent to which a user engages with the mHealth application deployed in the trial. There are several screens within the application that a user can view. Across all users we measure the 40^{th} percentile of number of screens viewed on day d. If user i views more than this percentile, we set their engagement level to 1, otherwise it is 0.

Exposure to treatment This variable captures the extent to which a user is treated, or the treatment dosage experienced by this user. Let D_i denote the exposure to treatment for user i. Whenever a message is delivered to a user's phone D_ii s updated. That is, if a message is delivered between time t and $t+1$, $D_{t+1} = \lambda D_t + 1$. If a message is not delivered, $D_{t+1} = \lambda D_t$. Here, we se λ according to data from HeartStepsV1 and initialize D to 0.

Acknowledgements This material is based upon work supported by: NIH/NIAAA R01AA23187 ,NIH/NIDA P50DA039838,NIH/NIBIB U54EB020404 and NIH/NCI U01CA229437. The views expressed in

this article are those of the authors and do not necessarily reflect the official position of the National Institutes of Health, or any other part of the U.S. Department of Health and Human Services.

Declarations

Institutional Review Board Approval The HeartSteps study discussed here was approved by the Kaiser Permanente Washington Region Institutional Review Board under IRB number 1257484-14.

References

Abeille, M., Lazaric, A., et al. (2017). Linear thompson sampling revisited. *Electronic Journal of Statistics, 11*(2), 5165–5197.

Agrawal, S., & Goyal, N. (2012). Analysis of thompson sampling for the multi-armed bandit problem. *In: Conference on Learning Theory*, pp 39–1.

Agrawal, S., & Goyal, N. (2013). Thompson sampling for contextual bandits with linear payoffs. *In: International Conference on Machine Learning*, pp 127–135.

Pedregosa, F., et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research, 12,* 2825–2830.

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

Bonilla, E.V., Chai, K.M., & Williams, C. (2008). Multi-task Gaussian process prediction. *In: Advances in neural information processing systems*, pp 153–160.

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

Brochu, E., Hoffman, M.W., & de Freitas, N. (2010). Portfolio allocation for Bayesian optimization. arXiv preprint arXiv:10095419.

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

Casella, G. (1985). An introduction to empirical Bayes data analysis. *The American Statistician, 39*(2), 83–87.

Cesa-Bianchi, N., Gentile, C., & Zappella, G. (2013). A gang of bandits. *In: Advances in Neural Information Processing Systems*, pp 737–745.

Cheung, W.C., Simchi-Levi, D., & Zhu, R. (2018). Learning to optimize under non-stationarity. arXiv preprint arXiv:181003024.

Chowdhury, S. R., & Gopalan, A. (2017). On kernelized multi-armed bandits. *International Conference on Machine Learning, 70,* 844–853.

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

Consolvo, S., McDonald, D.W., Toscos, T., Chen, M.Y., Froehlich, J., Harrison, B., Klasnja, P., LaMarca, A., LeGrand, L., & Libby, R., et al. (2008). Activity sensing in the wild: a field trial of ubifit garden. *In: Proceedings of the SIGCHI conference on human factors in computing systems*, pp 1797–1806.

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

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

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

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

Finn, C., Rajeswaran, A., Kakade, S., & Levine, S. (2019). Online meta-learning. arXiv preprint arXiv:190208438.

Forman, E. M., Kerrigan, S. G., Butryn, M. L., Juarascio, A. S., Manasse, S. M., Ontañón, S., et al. (2018). Can the artificial intelligence technique of reinforcement learning use continuously-monitored digital data to optimize treatment for weight loss? *Journal of behavioral medicine, 42*(2), 276–290.

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

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

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

Hamine, S., Gerth-Guyette, E., Faulx, D., Green, B. B., & Ginsburg, A. S. (2015). Impact of mhealth chronic disease management on treatment adherence and patient outcomes: a systematic review. *Journal of medical Internet research, 17*(2), e52.

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

Kim, B., Tewari, A. (2019). Near-optimal oracle-efficient algorithms for stationary and non-stationary stochastic linear bandits. arXiv preprint arXiv:191205695.

Kim, B., & Tewari, A. (2020). Randomized exploration for non-stationary stochastic linear bandits. *In: Conference on Uncertainty in Artificial Intelligence*, pp 71–80.

Klasnja, P., Hekler, E.B., Shiffman, S., Boruvka, A., Almirall, D., Tewari, A., & Murphy, S.A. (2015). Microrandomized trials: An experimental design for developing just-in-time adaptive interventions. *Health Psychology* 34(S):1220.

Klasnja, P., Smith, S., Seewald, N. J., Lee, A., Hall, K., Luers, B., et al. (2018). Efficacy of contextually tailored suggestions for physical activity: A micro-randomized optimization trial of heartsteps. *Annals of Behavioral Medicine, 53*(6), 573–582.

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

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

Lawrence, N.D., & Platt, J.C. (2004). Learning to learn with the informative vector machine. *In: International conference on Machine learning*, p 65.

Li, L., Chu, W., Langford, J., & Schapire, R.E. (2010). A contextual-bandit approach to personalized news article recommendation. *In: Proceedings of the Conference on World wide web*, pp 661–670.

Li, S., & Kar, P. (2015). Context-aware bandits. arXiv preprint arXiv:151003164.

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

Liao, P., Greenewald, K., Klasnja, P., & Murphy, S. (2020). Personalized heartsteps: A reinforcement learning algorithm for optimizing physical activity. *Proceedings of the Conference on Interactive, Mobile, Wearable and Ubiquitous Technologies, 4*(1), 1–22.

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

Morris, C. N. (1983). Parametric empirical Bayes inference: theory and applications. *Journal of the American statistical Association, 78*(381), 47–55.

Nagabandi, A., Finn, C., & Levine, S. (2018). Deep online learning via meta-learning: Continual adaptation for model-based rl. arXiv preprint arXiv:181207671.

Nahum-Shani, I., Smith, S. N., Spring, B. J., Collins, L. M., Witkiewitz, K., Tewari, A., & Murphy, S. A. (2017). 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).

Paredes, P., Gilad-Bachrach, R., Czerwinski, M., Roseway, A., Rowan, K., & Hernandez, J. (2014). Poptherapy: Coping with stress through pop-culture. *In: Conference on Pervasive Computing Technologies for Healthcare*, pp 109–117.

Qi, Y., Wu, Q., Wang, H., Tang, J., & Sun, M. (2018). Bandit learning with implicit feedback. *Advances in Neural Information Processing Systems, 31,* 7276–7286.

Qian, T., Klasnja, P., & Murphy, S.A. (2019). Linear mixed models under endogeneity: modeling sequential treatment effects with application to a mobile health study. arXiv preprint arXiv:190210861.

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

Rabbi, M., Philyaw-Kotov, M., Lee, J., Mansour, A., Dent, L., Wang, X., Cunningham, R., Bonar, E., Nahum-Shani, I., & Klasnja, P., et al. (2017). SARA: a mobile app to engage users in health data collection. *In: Joint Conference on Pervasive and Ubiquitous Computing and the International Symposium on Wearable Computers*, pp 781–789.

Raudenbush, S.W., & Bryk, A.S. (2002). Hierarchical linear models: Applications and data analysis methods, vol 1.

Russac, Y., Vernade, C., & Cappé, O. (2019). Weighted linear bandits for non-stationary environments. *In: Advances in Neural Information Processing Systems*, pp 12017–12026.

Russo, D., & Van Roy, B. (2014). Learning to optimize via posterior sampling. *Mathematics of Operations Research, 39*(4), 1221–1243.

Russo, D.J., Roy, B.V., Kazerouni, A., Osband, I., & Wen, Z. (2018). A tutorial on thompson sampling. *Foundations and Trends in Machine Learning* 11(1):1–96, https://doi.org/10.1561/2200000070.

Sæmundsson, S., Hofmann, K., & Deisenroth, M.P. (2018). Meta reinforcement learning with latent variable gaussian processes. arXiv preprint arXiv:180307551.

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

Srinivas, N., Krause, A., Kakade, S.M., & Seeger, M. (2009). Gaussian process optimization in the bandit setting: No regret and experimental design. *International Conference on Machine Learning* p 1015–1022.

Thompson, W. R. (1933). On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika, 25*(3/4), 285–294.

Vaswani, S., Schmidt, M., & Lakshmanan, L. (2017). Horde of bandits using Gaussian Markov random fields. *In: Artificial Intelligence and Statistics*, pp 690–699.

Wang, Y., & Khardon, R. (2012). Nonparametric Bayesian mixed-effect model: A sparse Gaussian process approach. arXiv preprint arXiv:12116653.

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

Williams, C. K., & Rasmussen, C. E. (2006). *Gaussian processes for machine learning* (Vol. 2). MA: MIT press Cambridge.

Xia, I. (2018). The price of personalization: An application of contextual bandits to mobile health. Senior thesis.

Yom-Tov, E., Feraru, G., Kozdoba, M., Mannor, S., Tennenholtz, M., & Hochberg, I. (2017). Encouraging physical activity in patients with diabetes: intervention using a reinforcement learning system. *Journal of medical Internet research, 19*(10), e338.

Zhao, P., Zhang, L., Jiang, Y., & Zhou, Z.H. (2020). A simple approach for non-stationary linear bandits. *In: Proceedings of the Conference on Artificial Intelligence and Statistics*, pp 746–755.

Zhou, M., Mintz, Y., Fukuoka, Y., Goldberg, K., Flowers, E., Kaminsky, P., Castillejo, A., & Aswani, A. (2018). Personalizing mobile fitness apps using reinforcement learning. *In: CEUR workshop proceedings*, vol 2068.

Zintgraf, L.M., Shiarlis, K., Kurin, V., Hofmann, K., & Whiteson, S. (2019). CAML: Fast context adaptation via meta-learning. *In: International Conference on Machine Learning*, pp 7693–7702.