DTU Library

Simulation Environment for Guiding the Design of Contextual Personalization Systems in the Context of Hearing Aids

Korzepa, MacIej; Petersen, Michael Kai; Larsen, Jakob Eg; Mørup, Morten

Published in:
Adjunct Publication of the 28th ACM Conference on User Modeling, Adaptation and Personalization

Link to article, DOI:
10.1145/3386392.3399291

Publication date:
2020

Document Version
Publisher's PDF, also known as Version of record

Citation (APA):
Korzepa, M., Petersen, M. K., Larsen, J. E., & Mørup, M. (2020). Simulation Environment for Guiding the Design of Contextual Personalization Systems in the Context of Hearing Aids. In *Adjunct Publication of the 28th ACM Conference on User Modeling, Adaptation and Personalization* (pp. 293-298). Association for Computing Machinery. https://doi.org/10.1145/3386392.3399291

Simulation Environment for Guiding the Design of Contextual Personalization Systems in the Context of Hearing Aids

Maciej Korzepa
mjko@dtu.dk
Technical University of Denmark
Kongens Lyngby, Denmark

Michael Kai Petersen
Eriksholm Research Centre, Demant
Snekkersten, Denmark

Jakob Eg Larsen
Morten Mørup
Technical University of Denmark
Kongens Lyngby, Denmark

ABSTRACT

Adjusting the settings of hearing aids in a clinic is challenging as the measured thresholds of audibility do not reflect many aspects of cognitive perception or the resulting differences in auditory preferences across different contexts. Online personalization systems have a potential to solve this problem, yet the lack of contextual user preference data constitutes a major obstacle in designing and implementing them. To address this challenge, we propose a simulation-based framework to inform and accelerate the development process of online contextual personalization systems in the context of hearing aids. We discuss how to model hearing aid users and context allowing partial observability, and propose how to generate plausible preference models using Gaussian Processes incorporating assumptions about the environment in a controlled way. Finally, on a simple example we demonstrate how an uncertainty-driven agent can efficiently learn from noisy user responses within the proposed framework. We believe that such simulated environments are vital for successful development of complex context-aware online recommender systems.

CCS CONCEPTS

• **Computing methodologies** → **Simulation environments**; *Gaussian processes*; • **Human-centered computing** → **Contextual design**; User models; • **Information systems** → *Recommender systems.*

KEYWORDS

contextual personalization; simulation environments; hearing aids

ACM Reference Format:
Maciej Korzepa, Michael Kai Petersen, Jakob Eg Larsen, and Morten Mørup. 2020. Simulation Environment for Guiding the Design of Contextual Personalization Systems in the Context of Hearing Aids. In *Adjunct Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization (UMAP '20 Adjunct), July 14–17, 2020, Genoa, Italy*. ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3386392.3399291

ACM ISBN 978-1-4503-7950-2/20/07…$15.00
https://doi.org/10.1145/3386392.3399291

1 INTRODUCTION

The increasing prevalence of hearing loss combined with limited clinical resources lead to users being dispensed hearing aids (HA) fitted purely based on audiograms which only measure the loss of audibility at different frequencies. However, as hearing loss is a complex distortion in auditory nerve activity patterns rather than just a loss of sensitivity [13], audiogram-based fitting cannot address differences among the hearing-impaired such as in speech understanding in noisy environments [14]. Moreover, apart from large differences *between* individuals, there is often lots of variation in preferences *within* individuals. For example, binnaural loudness perception may vary up to 30dB depending on the characteristics of the soundscape [16]. Similarly, depending on auditory intents in various daily situations and environments, users show preferences for very contrasting settings [11, 12]. These findings indicate that extensive contextual personalization is essential for the hearing impaired to minimize the risk of experiencing serious consequences of hearing loss related to e.g. well-being and cognitive load [2].

Multiple attempts have been made at optimizing hearing aid settings based on user feedback addressing different aspects such as fine-tuning frequency gain curve in static environments using Bayesian optimization [15], context-aware online preference learning using reinforcement learning (RL) [1] and learning shared preferences with hierarchical Bayesian models[5]. While all these aspects appear to be very important in designing systems for personalization of HA settings, there is, as of now, no system that incorporates all of them. When designing such a system, it is tempting to take a more holistic perspective and think of HA as an interactive, context-aware, online recommender system [17] that should continually adapt not only to users' setting preferences, but also to different characteristics of how users interact with the system, such as level of engagement, frequency of interactions or consistency of feedback, which can vary tremendously across users. A major obstacle in developing such a system is the lack of data that would guide its iterative design and evaluation. User studies are typically very time-consuming, costly, limited in size and carried out when the designer believes the prototype (which is developed without adequate data) is ready for testing. This renders the development of a complex personalization system very difficult.

A promising direction to facilitate development and evaluation of recommender systems in domains without adequate historical data is designing a simulated environment that incorporates relevant characteristics of the real one. Ie et al. [10] recently proposed RecSim, a framework for constructing such dynamic environments that allows for evaluation of RL-based recommendation agents. Using this framework, the goal of the modeler is to define the building blocks specifying in a probabilistic way what characteristics users consist of, how these characteristics contribute to liking specific recommended items and how they evolve over time. Importantly, the goal is not to create a completely realistic environment, but rather a tool that will allow to test agent's learning capabilities with respect to various assumptions the designer wants to incorporate about the environment such as partial observability of user or context features, noisiness of responses or complexity of preferences.

In this article, we extend this framework and adapt it to the HA domain. First, we introduce user-dependent context that, together with user characteristics, conditions user preferences. The context assumes partial observability which is crucial to evaluate how the inability to observe all the relevant context can potentially impede the agent's learning performance. Secondly, we suggest how to model context and user features in the HA domain using commonly available data. Finally, we present a Gaussian Process based model for sampling plausible contextual preferences whose complexity, dependence on observable and hidden features as well as degree of correlation between different users can be easily controlled with a set of hyperparameters. Even though the proposed approach is presented in the context of the HA domain, the generative preference model works with arbitrary context and user features and thus can be easily adapted to other domains. While modelling agents is not the focus of this paper, we also implement a simple agent and perform a simple simulation to demonstrate how agents can interact with the proposed environment and learn.

2 SIMULATION ENVIRONMENT

Hearings aids represent a domain in which users might have completely different preferences depending on the context they are in. Settings that work well for a user in one situation might totally fail in another one. While introducing context-awareness in a personalization system allows to distinguish between various situations and potentially offer optimal settings at the right moment, it also adds an extra layer of complexity and challenges to the problem. What if the observed context does not reflect user preferences well? Can the agent actually learn contextual preferences in an online setting? Can the agent benefit from correlations between contextual preferences of multiple users? What if users' contextual preferences are more complex than one expects? It is crucial for the designer of a recommendation agent to have some insight into its flexibility and performance under a range of different assumptions about the environment.

To facilitate informed development of agents, we borrow and adapt the RecSim simulation framework proposed by Ie et al. [10]. We present the modified diagram of the data flow in the framework in Fig. 1. A single cycle of contextual recommendation starts by sampling a user from a set of users generated for the simulation. The user is represented by a set of observable and hidden features according to the user model. Next, the context, also consisting of observable and hidden features, is sampled according to the user's context model (which might have a temporal structure). The agent observes the observable components of user and context features and based on them generates a recommendation with a number of settings within a given setting space. Then, the user generates a noisy response to the recommendation based on the internal contextual preference model. The agent observes the user's response

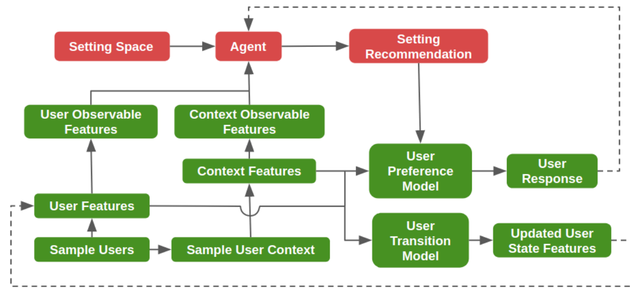


Figure 1: Data flow diagram of simulator for contextual preference learning based on RecSim [10].

and updates its internal recommendation model. Additionally, the state of the user changes according to the user's transition model (e.g. engagement might increase or decrease).

In this paper, we focus on the construction of user and context features in the HA domain and propose a model for generating contextual preference models in a controlled way. We start with a high-level overview of the proposed approach in section 2.1. In sections 2.2 and 2.3 we discuss how context and user features can be modeled in the HA domain, and define mechanisms that will allow control over generated preferences. The details of how we generate contextual preferences are specified in section 2.4. Section 2.5 relates user preferences to an actual response to agent's recommendations. Finally, in section 3 we present a simple simulation to demostrate how agents interact with the environment and conclude with a discussion in section 4.

2.1 Overview

We start by presenting an outline of the approach we propose to generate user preference models. We define a HA setting vector $\mathbf{s} \in \mathbb{R}^K$ (e.g. low-dimensional reparametrization of added gain at different frequencies) that we can control, and context vector $\mathbf{c} \in \mathbb{R}^D$ that fully explains user auditory needs at given instance of time. Our goal is to create for each user a function $(\mathbf{c}, \mathbf{s}) \mapsto f(\mathbf{c}, \mathbf{s}) \in \mathbb{R}$ that maps context and settings to a latent preference value which indicates the degree to which the user likes settings $\mathbf{s}$ in context $\mathbf{c}$. Additionally, we want to be able to induce correlations between functions for similar users and be able to evaluate the function efficiently for any $\mathbf{c}$ and $\mathbf{s}$. To model these functions we employ Gaussian Processes (GPs), a nonparametric class of models that enables straightforward control over complexity of the functions and correlations between them through kernels [18]. We first use a GP prior to sample preferred settings over a specified range of contexts to construct preference datasets that incorporate assumptions about correlations of user preferences and their complexity. We then define the latent preference of a user to be the posterior mean of another GP trained using the dataset specific to that user which allows us to evaluate their preference for any point in the complete domain of possible contexts and settings.

2.2 Context model

Nowadays HA users are not limited to elderly spending most of their time at home, but include many who are professionally and socially

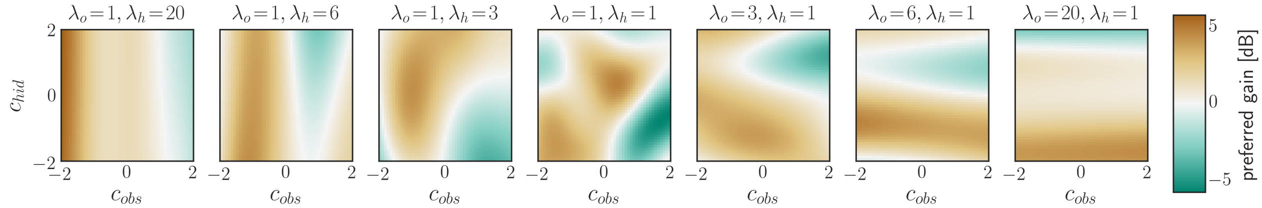


Figure 2: Varying the ratio of context lengthscale parameters λ_o and λ_h for observed and latent components allows us to sample user preferences with different assumptions about how well the observed context explains the actual user preferences.

active. Consequently, the environments and situations experienced by HA users are greatly diverse and so are their auditory needs. Korzepa et al. [11] provides an overview of various context types that might be crucial for identifying auditory preferences. They include characteristics of the acoustic scene at low level (e.g. sound pressure level, frequency composition) or high level (e.g. restaurant or office sounds, voice composition), activity (e.g. driving, exercising), location and time. The range of potentially important contexts for recognizing specific preferences is extremely wide and in practice, e.g. due to technical limitations or privacy concerns, it will never be possible to capture all the relevant context information. To take this into account in the simulation environment, we assume partial observability of the context, i.e. $\mathbf{c} = (\mathbf{c}_\mathrm{o}, \mathbf{c}_\mathrm{h})$ with $\mathbf{c}_\mathrm{o}$ and $\mathbf{c}_\mathrm{h}$ denoting the observable and hidden components of context $\mathbf{c}$.

Realistic HA user's context can be generated based on previously collected data and be user-dependent e.g. by modelling it as a Bayesian network [8]. In the proposed approach we assume that $\mathbf{c}$ is a low-dimensional representation obtained by dimensionality reduction of the actual context features e.g. by Principal Component Analysis, or Factor Analysis for Mixed Data if context features include categorical types. The hidden component of the context can be obtained by e.g. treating some of the observed features as hidden. To correlate preference in context space, we define the context kernel:

$$\kappa_c(\mathbf{c}, \mathbf{c}'; \boldsymbol{\lambda}^c) = \kappa_{\mathrm{RBF}}(\mathbf{c}_\mathrm{o}, \mathbf{c}'_\mathrm{o}; \lambda^c_\mathrm{o}) \cdot \kappa_{\mathrm{RBF}}(\mathbf{c}_\mathrm{h}, \mathbf{c}'_\mathrm{h}; \lambda^c_\mathrm{h}), \quad (1)$$

where $\kappa_{\mathrm{RBF}}(x, x', \lambda)$ denotes a radial basis function (RBF) kernel with lengthscale λ and unit output variance, in which the correlation between the modelled function falls with the increasing distance between the inputs (i.e. context).

In this paper, for the demonstration purposes, we do not assume any specific model on context $\mathbf{c}$ and simply define $\mathbf{c} = (c_{\mathrm{obs}}, c_{\mathrm{hid}})$ with $c_{\mathrm{obs}} \in [-2, 2]$ and $c_{\mathrm{hid}} \in [-2, 2]$ representing scalar observed and hidden context values. Moreover, we limit the settings to a single scalar value representing overall added gain.

By running simulations with different values of lengthscale parameters $\boldsymbol{\lambda}^c$, we are able to evaluate different aspects of agent's performance. One of the most imporant ones might be to measure how well the agent performs when user preferences can be fully explained by the observed context and how quickly its performance degrades with an increasing contribution of the hidden context. We can generate preferences with different degree of dependence on observed and hidden context by controlling ratio λ^c_h/λ^c_o as demonstrated in Fig. 2. Another important aspect is how complex preferences the agent is able to learn efficiently. We can generate preferences of different complexity by adjusting the magnitude of the lengthscale parameters as shown in Fig. 3.

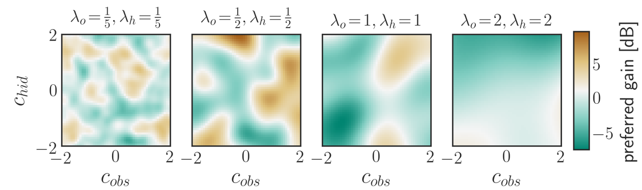


Figure 3: Varying context lengthscale parameters allows us to sample preferences with different degree of smoothness.

2.3 User model

While auditory preferences can vary greatly across users, not all users are completely different [4, 17]. Within a large population of users, it is natural that there are some groups of users who exhibit similar preferences, e.g. a preference towards enhanced brightness in speech environments [17]. We hypothesize that similarities in user preferences are the result of similarities in some characteristics that define users. These characteristics might include both observable and hidden features. A promising strategy to define observable features is to characterize user's hearing loss across two independent dimensions being audibility-related distortions (e.g. as specified by an audiogram) and non-audibility-related distortions characterizing reduction in binaural and temporal fine-structure processing abilities [19]. If such observable characteristics give rise to specific kind of preferences, knowing them would be crucial to alleviate cold start problem for new users when the agent has not collected any preference data about them yet.

Hidden features might relate to various cognitive aspects that are hard or impossible to measure. Hidden features can be modelled e.g. by treating some of the observed ones as hidden or constructing a Gaussian mixture (or a different) model that generates them.

We define user-specific feature vector $\mathbf{u}$ as a concatenation of observable and hidden user-specific feature vectors, $\mathbf{u}_o$ and $\mathbf{u}_h$ respectively, i.e. $\mathbf{u} = (\mathbf{u}_o, \mathbf{u}_h)$, and a distribution $p(\mathbf{u})$ that generates them. Through simulation, we expect to measure the impact that the presence of preferences induced by observable and hidden user characteristics has on the agent's learning performance. To be able to generate such preferences in a controlled way, we propose a kernel that is a weighted combination of RBF kernels with lengthscales $\boldsymbol{\lambda}^u = (\lambda^u_o, \lambda^u_h)$ operating on $\mathbf{u}_o$ and $\mathbf{u}_h$ respectively:

$$\kappa_u(\mathbf{u}, \mathbf{u}', \boldsymbol{\lambda}^u) = \theta \kappa_{\mathrm{RBF}}(\mathbf{u}_\mathrm{o}, \mathbf{u}'_\mathrm{o}; \lambda^u_o) + (1-\theta)\kappa_{\mathrm{RBF}}(\mathbf{u}_\mathrm{h}, \mathbf{u}'_\mathrm{h}; \lambda^u_h). \quad (2)$$

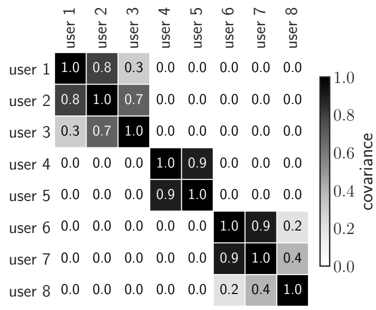


Figure 4: User kernel κ_u allows to model correlations between preferences of different users. Here we present the kernel matrix for a set of eight users for which we generate sample preferences as can be seen in Fig. 5.

Parameter θ controls the degree to what the preferences are induced by the observed and by the hidden features. For two users characterized by feature vectors $\mathbf{u}_i$ and $\mathbf{u}_j$, we expect their preferences to be identical if $\kappa_u(\mathbf{u}_i, \mathbf{u}_j) = 1$ and completely independent if $\kappa_u(\mathbf{u}_i, \mathbf{u}_j) = 0$.

In this paper, as we focus on showing how to generate correlated preferences based on similarities between users rather than using real audiological data to model the actual features, we arbitrarily construct observed feature vectors $\mathbf{u}_o$ and set θ to 1. We show the resulting kernel matrix shown in Fig. 4.

It is also important in a simulation to allow for behavioral and personal differences between users. HA users can differ in the extent of HA usage, engagement in the hearing loss treatment, familiarity with new techonologies or cognitive abilities to distinguish between differences in sound produced by HA. In this paper, due to space limitations, we do not include them in the simulation, but we get back to them in the discussion in section 4.

2.4 Preference model

Over a wide range of settings, there tend to exist settings that a user finds more effective than average and ones less effective than average. We refer to them as positive and negative preferences. As the task of an agent is to learn which settings are the effective ones, we focus on modeling positive preferences. We assume that given a fixed context $\mathbf{c}$ user's preference peaks around a single setting $\hat{\mathbf{s}}$ and falls towards the average preference away from it. This assumption seems plausible if the settings are parametrized such that larger distance between two settings corresponds to a bigger perceptual difference in their resulting HA output sounds.

We model preferences for M users whose feature vectors $\mathbf{u}$ we sample from the user generating distribution $p(\mathbf{u})$. For each user j, we aim to construct a training dataset $\mathcal{D}_j = \{(\mathbf{c}_i, \hat{\mathbf{s}}_i, f_H) | i = 1, \ldots, N\}$ that consists of N pairs indicating optimal setting $\hat{\mathbf{s}}_i$ for context $\mathbf{c}_i$ for which the user's latent preference f attains some high value f_H. We assume that individual dimensions s_k of setting $\mathbf{s}$ can be modelled independently. We define a multi-task GP [6] with $\hat{S}_k \in \mathbb{R}^M$ output where j-th output/task models preferred s_k for user j:

$$\hat{S}_k \sim \mathcal{GP}(\mathbf{0}, \delta \cdot \kappa_c(\mathbf{c}, \mathbf{c}'; \boldsymbol{\lambda}_s^c) \cdot \kappa_u(\mathbf{u}, \mathbf{u}'; \boldsymbol{\lambda}_s^u))), \qquad (3)$$

where $\delta \in \mathbb{R}^+$ controls the output scale, κ_c is the context kernel as defined in (1) with $\boldsymbol{\lambda}_s^c$ lengthscales and κ_u is the user kernel as defined in (2) with $\boldsymbol{\lambda}_s^u$ lengthscales. As the correlations in the inputs and correlations in the outputs are independent of each other, the resulting kernel matrix is Kronecker factored [6] which enables efficient computations. We construct a representative set C_s of N contexts $\mathbf{c}_i$ that covers the space of possible contexts well (we can do it e.g. by taking $R \gg N$ samples from $p(\mathbf{c})$ and selecting a subset of N such that the distance between any two contexts in the subset is maximized) and draw a sample $\hat{\mathrm{S}}_k \in \mathbb{R}^{N \times M}$ from the GP prior defined in (3) for each setting dimension k. Then, the optimal setting $\hat{\mathbf{s}}_i$ in context $\mathbf{c}_i$ for user j is defined as $\hat{\mathbf{s}}_i = ((\hat{\mathrm{S}}_0)_{ij}, \ldots, (\hat{\mathrm{S}}_K)_{ij})$. Having dataset $\mathcal{D}_j$, we define a GP prior on f_j:

$$f_j \sim \mathcal{GP}(m_j, \kappa_c(\mathbf{c}, \mathbf{c}'; \boldsymbol{\lambda}_{\text{pref}}^c) \cdot \kappa_{\text{RQ}}(\mathbf{s}, \mathbf{s}'; \lambda^s, \alpha)), \qquad (4)$$

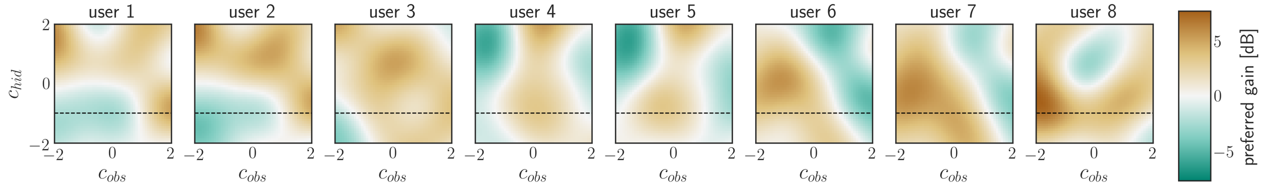
where m_j denotes user-dependent mean, κ_c is the context kernel as defined in (1) with $\boldsymbol{\lambda}_{\text{pref}}^c$ lengthscales and κ_{RQ} denotes a rational quadratic kernel with lengthscale λ^s and scale mixture parameter α which controls how quickly f_j decays towards the mean m_j away from the optimal settings. By performing noiseless GP regression, we can evaluate the GP posterior mean, $\mathbb{E}[f^* | \mathcal{D}_j, \mathbf{c}^*, \mathbf{s}^*]$ (which has a simple, well-known analytic solution [18]), at any context $\mathbf{c}^*$ for any setting $\mathbf{s}^*$, and we define it to be the latent preference of user j.

To visualize how the sampled settings and preferences reflect correlations between users, we construct a grid over c_{obs} and c_{hid}, use the user kernel matrix shown in Fig. 4, set δ to 10 and $\boldsymbol{\lambda}_s^c$ to $(1, 1)$ and draw a sample from the GP prior defined in (3). In Fig. 5a, we show the sampled preferred setting (gain) over the context grid for 8 users. Further, we set m_j to 0 for each j, $\boldsymbol{\lambda}_{\text{pref}}^c$ to $(0.15, 0.15)$, λ^s to 0.75 and α to 0.5, and generate preference models by performing GP regression on the previously sampled preferred setting dataset for each user. In Fig. 5b, we show how user preference for gain setting $s \in [-10, 10]$ changes over observed context c_{obs} with hidden context c_{hid} fixed to a specific value.

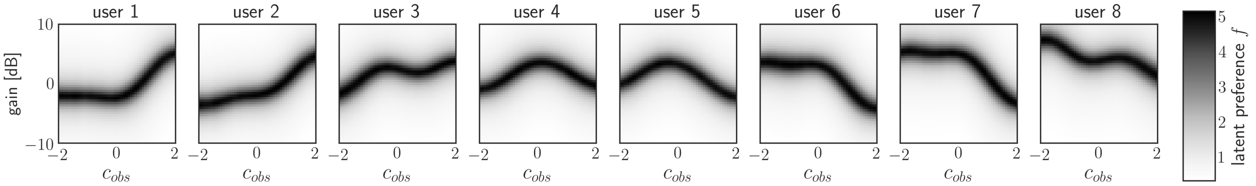
2.5 Response model

The agent learns from users' responses to offered recommendations. Even though users respond according to their latent preference model, their responses are inherently noisy (e.g. due to dependence on latent context or cognitive limitations to provide consistent feedback). The form of user response is typically decided by the system designer who takes into account suitability of a specific form from the users' perspective (e.g. what kind of response is more engaging or can be given quicker) and the agent's perspective (e.g. how efficiently it can learn). Some examples of response types proposed for personalization of hearing loss compensation include *binary* evaluations of sound quality for individual settings [20], or pair-wise comparisons of two settings using *continuous* response in $[0, 1]$ range (represented by a slider) where the deviation from the middle of the range indicates the degree of preference towards one or the other setting [15].

In our simulation, we assume users evaluate recommended settings independently by using a slider whose extreme positions,



(a) Preferred gain as a function of observed context c_{obs} and hidden context c_{hid}



(b) Gain preference at hidden context $c_{hid} = -1$

Figure 5: (a) Preferred gain setting for eight users sampled from a multi-task Gaussian Process prior with the task (user) covariance matrix shown in Fig. 4. (b) Latent preference of gain setting in $[-10, 10]$ range evaluated at $c_{\mathbf{hid}} = -1$ for the same users as in (a).

corresponding to values 0 and 1, indicate maximum dissatisfaction or satisfaction with a given setting. A natural choice of likelihood for user response y is then the Beta distribution as in [15]:

$$y \sim \text{Beta}(\nu \cdot \mu(f), \nu \cdot (1 - \mu(f))), \tag{5}$$

where $\mu(f)$ is a function mapping latent preference f to $[0, 1]$ range and defines the mean of the response, and ν controls the noise of the response (for $\nu \to \infty$, user response becomes noise-free i.e. $y \to \mu(f)$). We define the mapping μ as:

$$\mu(f) = \Phi(\sigma \cdot f), \tag{6}$$

where Φ denotes the cumulative standard normal distribution and σ controls the slope of the transition between negative and positive preference. Parameters ν and σ are typically user dependent (e.g. ν may depend on users' cognitive capabilities allowing to provide more or less consistent feedback). In a large-scale simulation, it would be reasonable to generate users with ν and σ sampled according to some prior distribution that we could construct based on our assumptions about it or historical data, if available. In our simulation, we simply set ν to 10 and σ to 0.4.

3 SIMULATIONS

In this paper, we do not focus on agent design and limit the scope of evaluation to learning preferences of an individual user. In our simulation we evaluate an agent based on Bayesian optimization. The agent maintains a surrogate GP model that, based on dataset $\mathcal{D}$ consisting of past interactions with the user, models user's posterior latent preference $p(f|\mathcal{D})$ with mean $\mu_f(\mathbf{c}, \mathbf{s})$ and variance $\sigma_f^2(\mathbf{c}, \mathbf{s})$ for context $\mathbf{c}$ and settings $\mathbf{s}$. When observing a new context $\mathbf{c}^*$, to decide on which setting $\mathbf{s}^*$ to recommend the agent optimizes Upper Confidence Bound (UCB) [7] over a given possible setting space $\mathcal{S}$:

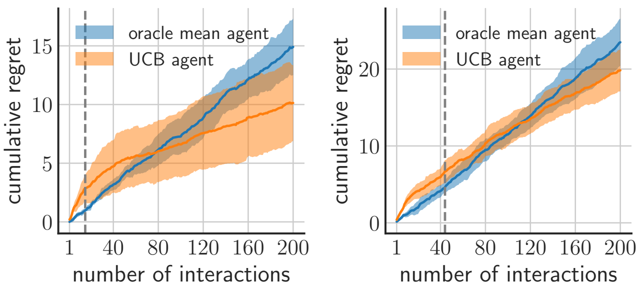
$$\mathbf{s}^* = \arg\max_{\mathbf{s} \in \mathcal{S}} \mu_f(\mathbf{c}^*, \mathbf{s}) + \beta \cdot \sigma_f(\mathbf{c}^*, \mathbf{s}), \tag{7}$$

where β controls the trade-off between exploration and exploitation. The agent's surrogate GP model uses Matern kernel and Beta likelihood parametrized as in (5) to map from latent function f to actual user response y. As the Beta likelihood is not conjugate with the GP prior, we use Variational Inference to infer the GP posterior and tune the hyperparametrs of the GP by maximizing the marginal likelihood of the data. We use GPyTorch [9] to implement the GP model with variational training scheme and employ BoTorch [3] to optimize the UCB aquisition function.

We compare the UCB agent to a simple baseline agent that always offers a constant setting $\hat{s}$ that maximizes average user preference over the distribution of user contexts $p(\mathbf{c})$ assumed in the simulation, i.e. $\hat{s} = \arg\max_s \mathbb{E}_{p(\mathbf{c})} f(\mathbf{c}, s)$. This represents the optimal recommendation *without* context-awareness. We refer to this agent as oracle mean agent. We measure the performance of each agent as the cumulative regret at interaction t given by $r_t = \sum_{i=1}^{t} \mu(\hat{f}_i) - y_i$, where $\hat{f}_i$ is the maximum user preference in context $\mathbf{c}_i$ observed at interaction i and y_i is the user response to the agent's recommendation at interaction i.

We run the simulation for user 4 (characterized by preferences shown in Fig. 5) and limit the number of interactions to $T = 200$. We consider two scenarios: full observability with hidden context c_{hid} fixed to 1 and partial observability with hidden context sampled uniformly from $[-2, 2]$ range at each interaction. In both scenarios, the observed context c_{obs} is sampled uniformly (and independently of the hidden context) from $[-2, 2]$ range. For UCB agent we set $\beta = 1$ and start with 5 random setting recommendations before the agent switches to its standard UCB acquisition. UCB agent retrains its surrogate GP model after each aquisition. We repeat the simulation 5 times for both scenarios with a different sample of T contexts.

We show the cumulative regret incurred by the agents in Fig. 6. In the full observability scenario, the UCB agent starts incurring consistently lower regrets than the oracle mean agent already after 15 interactions and finishes the simulation with 33% lower cumulative regret. In the partial observability scenario, the presence of hidden context greatly impedes the UCB agent's performance, but still it is able to take advatange of the observed context and perform better than oracle mean agent after 44 interactions. In both scenarios, one can clearly see the exploration and exploitation phases of the UCB agent.



(a) full observability **(b) partial observability**

Figure 6: Cumulative regret in full (a) and partial (b) observability scenarios averaged over 5 runs. The dashed line indicates the point at which the UCB agent starts incurring consistently lower regret than the oracle mean agent .

4 DISCUSSION

In our simulation we evaluated a simple Bayesian optimization agent in a single user environment. In practice, we are interested in simulating personalization on a scale of thousands of users using agents that also discover and exploit similarities between users to boost the learning speed. With a large pool of users, we can model the diversity of contextual preferences, as well as human factors such as cognitive perception or motivation. If users differ in the ability of providing consistent feedback, the agent might dynamically optimize its interventions. For users giving very inconsistent responses, the agent might learn to offer more contrasting settings and ask for a binary response to compensate for a high level of noise. Conversely, the agent might provide more fine-grained recommendations if user responses are consistent. Similarly, the frequency and timing of interventions may impact the users' motivation to interact with the agent. An intelligent agent should learn dynamically when and how to interact with users to keep them motivated over a long period of time. Simulation is a perfect tool to evaluate agents' capabilities of adaptation to such human factors and dynamic behaviors.

The closer the simulation environment is to the actual one, the less difference we expect between how well agents perform in simulation and in real world. When deploying the developed system for the first time, it is very likely that due to inaccurate assumptions the performance of the agent will be significantly worse than in the simulation. It is therefore reasonable to start with an agent that operates in a simpler space of settings and observed context than intended in the long run. Even if the first version fails to reach satisfactory performance, valuable insights about the characteristics of user preferences and interactions are gathered and can be used to revise the assumptions implemented in the simulator so that an improved agent can be developed. This creates a self-reinforcing loop - the data collected by agents improves the simulation which in turn leads to better performing agents.

The proposed framework allows to model contextual preferences considering both observable and latent aspects of cognitive perception in a flexible way. Simulation environments like this may be vital to shape new human-centered adaptive personalization systems in domains such as hearing healthcare where historical data is very limited and where (often dynamic) human factors play an important role in the personalization process.

REFERENCES

[1] Gabriel Aldaz, Sunil Puria, and Larry J. Leifer. 2016. Smartphone-based system for learning and inferring hearing aid settings. In *Journal of the American Academy of Audiology*, Vol. 27. American Academy of Audiology, 732–749. https://doi.org/10.3766/jaaa.15099
[2] Stig Arlinger. 2003. Negative consequences of uncorrected hearing loss-A review The n200 project View project. *Article in International Journal of Audiology* (2003). https://doi.org/10.3109/14992020309074639
[3] Maximilian Balandat, Brian Karrer, Daniel R Jiang, Samuel Daulton, Benjamin Letham, Andrew Gordon Wilson, and Eytan Bakshy. [n.d.]. *BOTORCH: PROGRAMMABLE BAYESIAN OPTIMIZATION IN PYTORCH.* Technical Report.
[4] Laura Winther Balling, Niels Søgaard Jensen, Oliver Townend, and Wendy Switalski. 2020. SoundSense Learn: When average is not enough. *Hearing Review* 27(2) (2020).
[5] Adriana Birlutiu, Perry Groot, and Tom Heskes. 2010. Multi-task preference learning with an application to hearing aid personalization. (2010). https://doi.org/10.1016/j.neucom.2009.11.025
[6] Edwin V Bonilla, Ming A Chai, and Christopher K I Williams. [n.d.]. *Multi-task Gaussian Process Prediction.* Technical Report.
[7] Eric Brochu, Vlad M Cora, and Nando De Freitas. 2010. *A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning.* Technical Report.
[8] Jeppe H. Christensen, Niels H. Pontoppidan, Rikke Rossing, Marco Anisetti, Doris Eva Bamiou, George Spanoudakis, Louisa Murdin, Thanos Bibas, Dimitris Kikidiks, Nikos Dimakopoulos, Giorgos Giotis, and Apostolos Ecomomou. 2019. Fully Synthetic Longitudinal Real-World Data From Hearing Aid Wearers for Public Health Policy Modeling. *Frontiers in Neuroscience* 13 (8 2019), 850. https://doi.org/10.3389/fnins.2019.00850
[9] Jacob R Gardner, Geoff Pleiss, David Bindel, Kilian Q Weinberger, and Andrew Gordon Wilson. [n.d.]. *GPyTorch: Blackbox Matrix-Matrix Gaussian Process Inference with GPU Acceleration.* Technical Report.
[10] Eugene Ie, Chih-Wei Hsu, Martin Mladenov, Vihan Jain, Sanmit Narvekar, Jing Wang, Rui Wu, and Craig Boutilier. 2019. *RECSIM: A Configurable Simulation Platform for Recommender Systems.* Technical Report.
[11] Maciej Jan Korzepa, Benjamin Johansen, Michael Kai Petersen, Jan Larsen, Jakob Eg Larsen, and Niels Henrik Pontoppidan. 2018. Modeling user intents as context in smartphone-connected hearing aids. In *UMAP 2018 - Adjunct Publication of the 26th Conference on User Modeling, Adaptation and Personalization.* Association for Computing Machinery, Inc, 151–155. https://doi.org/10.1145/3213586.3226211
[12] M.J. Korzepa, B. Johansen, M.K. Petersen, J. Larsen, J.E. Larsen, and N.H. Pontoppidan. 2018. Learning preferences and soundscapes for augmented hearing. In *CEUR Workshop Proceedings*, Vol. 2068.
[13] Nicholas A Lesica. 2018. Why Do Hearing Aids Fail to Restore Normal Auditory Perception? *Trends in Neurosciences* 41 (2018), 174–185. https://doi.org/10.1016/j.tins.2018.01.008
[14] Enrique A. Lopez-Poveda. 2014. Why do I hear but not understand? Stochastic undersampling as a model of degraded neural encoding of speech. , 348 pages. https://doi.org/10.3389/fnins.2014.00348
[15] Jens Brehm Bagger Nielsen, Jakob Nielsen, and Jan Larsen. 2015. Perception-based personalization of hearing aids using gaussian processes and active learning. *IEEE/ACM Transactions on Audio Speech and Language Processing* 23, 1 (1 2015), 162–173. https://doi.org/10.1109/TASLP.2014.2377581
[16] Dirk Oetting, Volker Hohmann, Jens E. Appell, Birger Kollmeier, and Stephan D. Ewert. 2018. Restoring perceived loudness for listeners with hearing loss. *Ear and Hearing* 39, 4 (2018), 664–678. https://doi.org/10.1097/AUD.0000000000000521
[17] Alessandro Pasta, Michael Kai Petersen, Kasper Juul Jensen, and Jakob Eg Larsen. 2019. Rethinking Hearing Aids as Recommender Systems. *CEUR Workshop Proceedings* 2439 (2019), 11–17.
[18] Carl Edward Rasmussen. 2006. Gaussian processes for machine learning. MIT Press.
[19] Raul Sanchez Lopez, Federica Bianchi, Michal Fereczkowski, Sébastien Santurette, and Torsten Dau. 2018. Data-Driven Approach for Auditory Profiling and Characterization of Individual Hearing Loss. *Trends in Hearing* 22 (1 2018), 12. https://doi.org/10.1177/2331216518807400
[20] Thijs Van De Laar and Bert De Vries. 2016. A Probabilistic Modeling Approach to Hearing Loss Compensation. (2016). https://doi.org/10.1109/TASLP.2016.2599275