



On preference learning based on sequential Bayesian optimization with pairwise comparison

Tanya Ignatenko^{a,*}, Kirill Kondrashov^a, Marco Cox^a, Bert de Vries^{a,b}

^a GN Advanced Science, John F Kennedylaan 2, 5612 AB Eindhoven, the Netherlands

^b Department of Electrical Engineering, Eindhoven University of Technology, 5612 AZ Eindhoven, the Netherlands

ARTICLE INFO

Keywords:

Preference learning
KL-divergence
Universal prediction
Bayesian inference
Intelligent agents

ABSTRACT

User preference learning is generally a hard problem. Individual preferences are typically unknown even to users themselves, while the space of choices is infinite. Here we study user preference learning from information-theoretic perspective. We model preference learning as a system with two interacting sub-systems, one representing a user with his/her preferences and another one representing an agent that has to learn these preferences. The user with his/her behavior is modeled by a parametric preference function. To efficiently learn the preferences and reduce search space quickly, we propose the agent that interacts with the user to collect the most informative data for learning. The agent presents two proposals to the user for evaluation, and the user rates them based on his/her preference function. We show that the optimum agent strategy for data collection and preference learning is a result of maximin optimization of the normalized weighted Kullback-Leibler (KL) divergence between true and agent-assigned predictive user response distributions. The resulting value of the KL-divergence, which we also call of a remaining system uncertainty (RSU), provides an efficient performance metric in the absence of the ground truth. This metric characterizes how well the agent can predict user and, thus, the quality of the underlying learned user (preference) model. Our proposed agent comprises sequential mechanisms for user model inference and proposal generation. To infer the user model (preference function), Bayesian approximate inference is used in the agent. The data collection strategy is to generate proposals, responses to which help resolving uncertainty associated with prediction of the user responses the most. The efficiency of our approach is validated by numerical simulations. Also a real-life example of preference learning application is provided.

1. Introduction

In this work, we study the problem of preference learning. We start with a motivating example presenting a preference learning problem in the context of the fitting a hearing aid (HA). This example allows us to set the scene for the preference learning problem and reason about the properties and requirements of the corresponding solution.

Motivating Example: Fitting and tuning of HAs has always been considered a tedious task of healthcare professionals (HCPs). Traditional approaches for fitting HA parameters rely on compensation of a user's hearing loss, based on audiograms, by applying rules such as, e.g., NAL-NL1 or NAL-NL2 [1]. These rules, however, do not take into account specific user preferences. Tuning of

* Corresponding author.

E-mail addresses: tignatenko@gn.com (T. Ignatenko), kkondrashov@gn.com (K. Kondrashov), mail@marcocox.com (M. Cox), bdevries@gn.com (B. de Vries).

<https://doi.org/10.1016/j.artint.2025.104400>

Received 18 January 2022; Received in revised form 1 July 2023; Accepted 3 August 2025

the HA parameters would happen at some later moment of time, at an HCP's office, when the user shares some qualitative feedback about his/her experience of using the HAs with the HCP. The tuning result heavily depends on the HCP's experience and expertise levels, and the degree to which the user feedback can be related to the required tuning. In practice, due to generic fitting and tuning procedures, HA users remain not fully satisfied with the results that leads to recurrent visits to HCPs or less frequent HA use. This gives a rise to the personalization problem of optimal HA parameter settings. It is desirable to empower the user to take direct decisions and have direct impact on the tuning process by creating an automated system for learning user preferences in an efficient and minimally obtrusive way.

In the HA example, equalizer tuning, simulations, rendering and other similar applications, the result of parameter tuning cannot be objectively evaluated: it depends on an individual perceptual subjective evaluation. Individual preferences are often unknown even to the users themselves and might require infinite number of trying different options to discover them. Moreover, from the user perspective, the relation between the tuning process and its outcome is not straightforward. Indeed, there are typically many interconnected parameters that have a nonlinear relationship with the final result. Therefore, in this work, we study a problem of efficient automated search for optimal parameters or, more generally, user preference by incorporating a user feedback into the learning cycle. From psychological prospective, see, e.g., [2], people find it easier to compare two options and indicate their preference for one of them, rather than to rate options according to a numeric scale. Therefore, to lower the burden on the user and guarantee reliability of the feedback, we consider the setting, where the user is required to evaluate a sequence of pairs of proposals, by stating his/her¹ choice for one of them.

The problem of preference learning has been an active research topic in the past years. Brochu et al. [3] studied active preference learning with Gaussian processes used to model preference function and expected improvement criterion as acquisition function for efficient data collection. A good overview of preference learning using Gaussian processes and further references are presented in [4]. González et al. [5] studied the problem of preferential Bayesian optimization of a black-box function with duelling bandits. In their work an unknown function is also modeled using Gaussian processes and considered acquisition strategies included expected improvement and duelling-Thompson sampling. Petrus et al. [6] studied the preference learning problem by incorporating expert knowledge, where the feedback provided to the agent is optimal projection information provided by the experts, and thus is purely instructive. Another direction of preference function modeling was taken in [7], where parametric user preference models were proposed. In their work, Thompson sampling together with Bayesian approximate inference was used to learn the user preferences and cumulative value of the preference function was used to assess the learning performance. It was shown that parametric approach could outperform preference function models based on Gaussian processes. Remarkably, in situations with preference learning, the optimal user preferences as well as the corresponding value of the preference function are unknown. Therefore, in the approaches based on the expected improvements, the current estimate of the user preference function for the optimal preference is used as a ground truth, which need not be close to the actual user preference. Hence, the procedure does not guarantee convergence to the true preference. Duelling approaches only take into account the second order statistics for defining acquisition strategies. Moreover, performance metrics such as expected improvement, valuation function or Copeland score do not guarantee convergence to the optimum.

In Bayesian optimization literature, other alternatives for acquisition functions were proposed that included entropy search (ES) [8] and predictive entropy search (PES) [9] approaches. ES-based acquisition function, however, relies on the entropy reduction relative to the uniform distribution and thus does not guarantee efficient exploration of the search space and correct localization of the user preferences. PES approach aims at exploring the data points that provide the highest information gain about location of the global maximum of some unknown function. Similar to ES and PES approaches, we study the problem of the preference learning from the information-theoretic perspective. While our approach results into the acquisition function similar in form to the PES one, it is conceptually very different. We provide theoretical foundation for the optimal agent strategy that combines both preference learning and data acquisition, and propose corresponding performance measure. Instead of using estimates of unobservable preference function values, we concentrate on the information provided in the user responses, in order to steer the learning process. The proposed performance measure can be efficiently used to evaluate the learning performance in the absence of true user preference information.

1.1. Summary of our contribution

The goal of this work is to design a Bayesian optimizer for user preference learning from binary choices. This optimizer has to find the optimal preferences of the user, given binary user responses, in as few trials as possible. Binary user responses correspond to user indication of his preference for one of the two presented choices. Our approach is based on a parametric user preference model, proposed in [7]. In this case, the optimal preference can be expressed as the location of the maximum of the latent user preference function that we need to infer. We take an information-theoretic point of view to the problem. We model the preference learning system as two interacting sub-systems, where one sub-system represents the user with his preferences and the other represents an agent that probes the user, such that it can efficiently learn the user preferences, see Fig. 1. The agent strategy is based on sequential reduction of uncertainty about the user after each interaction between the user and the agent. We show that the optimal strategy is the result of maximin optimization of the normalized weighted Kullback-Leibler (KL) divergence between true and agent-assigned predictive user response distributions. This approach results into a theoretically-grounded acquisition function. Moreover, we use the

¹ Later in the text, for compactness, we use "his" as a gender-neutral pronoun associated with a "user".

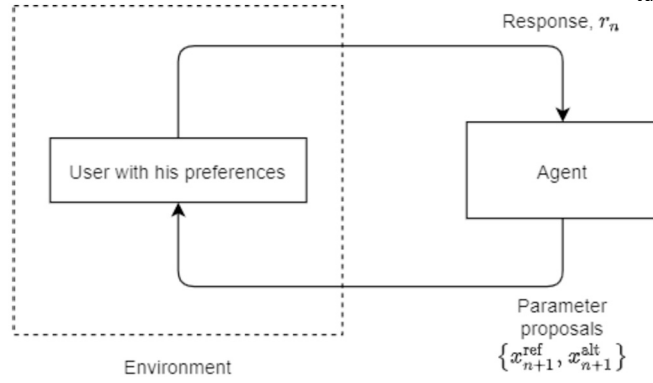


Fig. 1. A preference learning system, as two interacting sub-systems, i.e., a user and an agent. For each interaction, the agent generates two proposals. The user evaluates these two proposals, based on his preferences, and provides the ranking of the presented proposals (binary response).

value of the optimized KL-divergence as a metric that characterizes the quality of the learned preference and allows us to monitor the learning process and its convergence. This metric does not require the knowledge of the true preference. The optimal agent strategy and the metric are derived based on the concepts of universal prediction, see, e.g., [10]. Finally, our learning procedure is sequential, sometimes referred to as online, in the literature. The advantage of this approach is two-fold: firstly, we do not need to store all data required for the model update, and, secondly, this approach allows capturing the dynamics of (possibly) changing user preference.

Notations. In the rest of the text, we denote random variables by capital letters, R , realizations or observations by small letters, r , and vectors by bold letters, \mathbf{r} . Moreover, sequences are denoted by r_1^n and $\{a, b\}_1^n$, and defined as $r_1^n = \{r_1, r_2, \dots, r_n\}$ and $\{a, b\}_1^n = \{\{a_1, b_1\}, \{a_2, b_2\}, \dots, \{a_n, b_n\}\}$, respectively.

2. User and preference function modeling

We start with specifying the user model, which represents a probabilistic description of user responses, when comparing two proposals. The proposals are expressed as sets of parameters defining an outcome that user evaluates. E.g., in the motivating example, parameters are represented by a set of HA parameters that are used by a HA device to process sound, which, in its turn, is evaluated by a user. Thus, parameters are latent random variables in the models we propose. Throughout the text, we use parameter proposals, parameters and proposals interchangeably, and it will be clear from the context if proposals refer to the parameters or outcome of parameter application. Integral parts of the model include a user preference function and a distribution of the user responses to the presented choices of parameters.

Consider a vector of parameters, defined on a D -dimensional continuous compact surface. In particular, parameters \mathbf{x} are defined on a D -dimensional hyper-cube, i.e., $\mathbf{x} \in [0, 1]^D$. We can always guarantee this by normalizing the actual parameters by their physical range. Our goal is to find optimal values of parameters, denoted by θ , corresponding to the preference of a particular user. Here we follow the model setting of [7].

The user model is based on two assumptions: (1) a user preference exists and is described by a unimodal preference function $f(\cdot)$; and (2) a user can be uncertain about his preference, which can be expressed in terms of an additive random variable. In general, the user preference function is unknown. We assume this latent preference function $f(\mathbf{x}; \theta, \Lambda)$ to be a parametric function of parameters $\mathbf{x} \in [0, 1]^D$ with known form but unknown shape. This shape is characterized by *tuning parameters* $\theta \in [0, 1]^D$ and $\Lambda = \text{diag}([\lambda_1, \lambda_2, \dots, \lambda_D])$, $\lambda_d \in \mathbb{R}^+$, $d = 1, 2, \dots, D$. In particular, the user preference function is specified by the following function

$$f(\mathbf{x}; \theta, \Lambda) = -\sqrt{(\mathbf{x} - \theta)^T \Lambda (\mathbf{x} - \theta)}. \quad (1)$$

Note that the tuning parameters correspond to the location of the function optimum, θ , that is, in fact, the optimal parameters corresponding to the user preference; and to the spread around this optimum, Λ , that characterizes user sensitivity to parameter changes, respectively. The tuning parameters are user-specific and need to be learned. The form of this preference function has an attractive property that it allows for quick narrowing of the search space, see [7].

Next, in order to model user preference uncertainty, we define a value function $u(\cdot)$ as

$$u(\mathbf{x}) = f(\mathbf{x}; \theta, \Lambda) + \varepsilon, \quad (2)$$

where the user uncertainty error ε is assumed to have the standard Gaussian distribution, i.e.,

$$\varepsilon \sim \mathcal{N}(0, 1). \quad (3)$$

The choice for this assumption is further discussed after we introduce the notion of the user response.

Consider now a user response to a pairwise comparison of parameters from a trial, defined by a pair $\{\mathbf{x}^{\text{ref}}, \mathbf{x}^{\text{alt}}\}$, where \mathbf{x}^{ref} and \mathbf{x}^{alt} are the so-called *reference and alternative parameter proposals*, respectively. We call reference proposal \mathbf{x}^{ref} to be the highest rated proposal from the previous trial and this situation is challenged by the alternative proposal \mathbf{x}^{alt} , in the given trial. The user response depends on his preference for the presented parameters, which is given by the value function for these parameters. The user response, R , takes values from $\{0, 1\}$, where response 1 to a trial $\{\mathbf{x}^{\text{ref}}, \mathbf{x}^{\text{alt}}\}$ indicates that parameter \mathbf{x}^{alt} is preferred over or is as good as \mathbf{x}^{ref} , and 0, otherwise, i.e.,

$$R \stackrel{\Delta}{=} \begin{cases} 1, & \text{if } \mathbf{x}^{\text{alt}} \geq \mathbf{x}^{\text{ref}} \\ 0, & \text{if } \mathbf{x}^{\text{ref}} > \mathbf{x}^{\text{alt}} \end{cases}, \quad (4)$$

here \geq denotes preferred or equivalent to, and $>$ denotes strict preference.

Now, the probability of a positive user response, i.e., that R takes on value 1, in a trial $\{\mathbf{x}^{\text{ref}}, \mathbf{x}^{\text{alt}}\}$, is given by

$$\begin{aligned} \Pr\{R = 1 \mid \{\mathbf{x}^{\text{ref}}, \mathbf{x}^{\text{alt}}\}, f\} &= \Pr\{u(\mathbf{x}^{\text{alt}}) \geq u(\mathbf{x}^{\text{ref}})\} \\ &= \Pr\{f(\mathbf{x}^{\text{alt}}; \theta, \Lambda) + \varepsilon' \geq f(\mathbf{x}^{\text{ref}}; \theta, \Lambda) + \varepsilon\} \\ &= \Pr\{\varepsilon - \varepsilon' \leq f(\mathbf{x}^{\text{alt}}; \theta, \Lambda) - f(\mathbf{x}^{\text{ref}}; \theta, \Lambda)\} \\ &= \Phi\left(\frac{f(\mathbf{x}^{\text{alt}}; \theta, \Lambda) - f(\mathbf{x}^{\text{ref}}; \theta, \Lambda)}{\sqrt{2}}\right) \\ &= \Phi(f(\mathbf{x}^{\text{alt}}; \theta, \Lambda') - f(\mathbf{x}^{\text{ref}}; \theta, \Lambda')), \end{aligned} \quad (5)$$

where $\Phi(\cdot)$ is the cumulative distribution function (CDF) of the standard Gaussian distribution, and $\Lambda' = \frac{\Lambda}{2}$. Thus, R is a Bernoulli random variable with parameter $\Phi(f(\mathbf{x}^{\text{alt}}; \theta, \Lambda') - f(\mathbf{x}^{\text{ref}}; \theta, \Lambda'))$. Therefore, given the user preference and response model, defined by (1)-(5), the probability mass function (PMF) of a user response to a trial $\{\mathbf{x}^{\text{ref}}, \mathbf{x}^{\text{alt}}\}$ is given by

$$\begin{aligned} P(r \mid \{\mathbf{x}^{\text{ref}}, \mathbf{x}^{\text{alt}}\}, f) &= \Phi(f(\mathbf{x}^{\text{alt}}; \theta, \Lambda') - f(\mathbf{x}^{\text{ref}}; \theta, \Lambda'))^r \cdot (1 - \Phi(f(\mathbf{x}^{\text{alt}}; \theta, \Lambda') - f(\mathbf{x}^{\text{ref}}; \theta, \Lambda')))^{1-r} \\ &= \Phi(f(\mathbf{x}^{\text{alt}}; \theta, \Lambda') - f(\mathbf{x}^{\text{ref}}; \theta, \Lambda'))^r \cdot \Phi(f(\mathbf{x}^{\text{ref}}; \theta, \Lambda') - f(\mathbf{x}^{\text{alt}}; \theta, \Lambda'))^{1-r}. \end{aligned} \quad (6)$$

This user response model, which relates binary observations to the continuous latent function, is also known as the Thurstone-Mosteller law of comparative judgment [11]. In statistics, it is called the binomial-probit regression model.

Remark 2.1. Note that we can assume unit-variance of the user uncertainty error because other values can be absorbed by the preference function as a scaling factor in the user sensitivity. Indeed, when $\varepsilon \sim \mathcal{N}(0, \beta^{-1})$ the user response model becomes

$$\begin{aligned} P(r \mid \{\mathbf{x}^{\text{ref}}, \mathbf{x}^{\text{alt}}\}, f) &= \Phi\left(\frac{f(\mathbf{x}^{\text{alt}}; \theta, \Lambda) - f(\mathbf{x}^{\text{ref}}; \theta, \Lambda)}{\sqrt{2}\beta^{-1}}\right)^r \cdot \Phi\left(\frac{f(\mathbf{x}^{\text{ref}}; \theta, \Lambda) - f(\mathbf{x}^{\text{alt}}; \theta, \Lambda)}{\sqrt{2}\beta^{-1}}\right)^{1-r} \\ &= \Phi(f(\mathbf{x}^{\text{alt}}; \theta, \Lambda'') - f(\mathbf{x}^{\text{ref}}; \theta, \Lambda''))^r \cdot \Phi(f(\mathbf{x}^{\text{ref}}; \theta, \Lambda'') - f(\mathbf{x}^{\text{alt}}; \theta, \Lambda''))^{1-r}, \end{aligned}$$

where $\Lambda'' = \frac{\beta^2 \cdot \Lambda}{2}$.

Remark 2.2. Observe that, in this model, we can re-define Λ as Λ' or Λ'' . Since Λ is an unknown parameter, this scaling can be applied without loss of generality. However, care need to be taken while setting the priors for this parameter, taking into account physical interpretation of parameters in a specific application and thus possibly also re-scaling hyper parameters of the prior distributions.

The following definition summarizes our user model.

Definition 2.1 (User Model). A user model is defined by the PMF of the user response r to a trial $\{\mathbf{x}^{\text{ref}}, \mathbf{x}^{\text{alt}}\}$ and is given by

$$P(r \mid \{\mathbf{x}^{\text{ref}}, \mathbf{x}^{\text{alt}}\}, f) = \Phi(f(\mathbf{x}^{\text{alt}}; \theta, \Lambda') - f(\mathbf{x}^{\text{ref}}; \theta, \Lambda'))^r \cdot \Phi(f(\mathbf{x}^{\text{ref}}; \theta, \Lambda') - f(\mathbf{x}^{\text{alt}}; \theta, \Lambda'))^{1-r}, \quad (7)$$

where $\Phi(\cdot)$ is the CDF of the standard normal distribution. Here $f(\cdot; \cdot)$ is a user preference function, given by

$$f(\mathbf{x}; \theta, \Lambda) = -\sqrt{(\mathbf{x} - \theta)^T \Lambda (\mathbf{x} - \theta)}, \quad (8)$$

the user response relation to a trial $\{\mathbf{x}^{\text{ref}}, \mathbf{x}^{\text{alt}}\}$ is defined by

$$R \stackrel{\Delta}{=} \begin{cases} 1, & \text{if } \mathbf{x}^{\text{alt}} \geq \mathbf{x}^{\text{ref}} \\ 0, & \text{if } \mathbf{x}^{\text{ref}} > \mathbf{x}^{\text{alt}} \end{cases}, \quad (9)$$

and the user uncertainty about his response to a given trial is characterized by the standard Gaussian distribution:

$$\varepsilon \sim \mathcal{N}(0, 1).$$

Observe that the user model, defined as above, is a parametric model of the user preferences and sensitivity. Our goal is to estimate the user preference parameter θ , based on a sequence of trials and corresponding user responses. This problem can be seen as a problem of finding the optimum of the latent user preference function, and thus requires learning the function or, equivalently, its parameters. Note, however, that sensitivity parameter Λ is, in fact, a nuisance parameter, since our primary goal is learning the location of the optimum. Moreover, since we need to learn user preferences in as few trials as possible, this learning problem also relates to the problem of active learning. There, the goal is to learn a model efficiently by selecting the most relevant data (trials in our setting). Given unimodality of the preference function, it is also desirable to construct trials such that in each given trial, an alternative proposal is better than a reference proposal with high probability. The resulting procedure is then similar in flavor to gradient ascent.

In the next section, we define the inference strategy for user preference learning, but, first, we specify the full model for our preference learning system and its properties.

3. Preference learning system and inference strategy

3.1. Parameter transformation

Before going into the model specification, we introduce useful parameter transformation and define the corresponding parameter vector:

$$\phi = \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_D \\ \gamma_1 \\ \vdots \\ \gamma_D \end{bmatrix}, \quad (11)$$

where

$$\alpha_d = \Phi^{-1}(\theta_d), \quad (12)$$

$$\gamma_d = \ln(\lambda_d), \quad d = 1, 2, \dots, D, \quad (13)$$

and $\Phi^{-1}(\cdot)$ is the inverse of the CDF of the standard Gaussian distribution. The latter non-linear monotonic transformation guarantees that the values of parameters θ are constrained to the $[0, 1]^D$ hyper-cube during the learning process, while we operate in the transformed continuous parameter space. From now on, we also assume that trial proposals are in the same transformed domain as the parameters. Similarly, transformation of the user sensitivity helps to constrain this parameter to the positive real subspace. Further, we re-define the preference function with this transformed parameter vector as

$$f(\mathbf{x}; \phi) \triangleq f(\Phi(\mathbf{x}); \Phi(\alpha), \exp(\Gamma)), \quad (14)$$

where the transformations are applied component-wise to α and \mathbf{x} , thus, e.g., $\Phi(\alpha)$ denotes a vector $\Phi(\alpha) = \{\Phi(\alpha_1), \Phi(\alpha_2), \dots, \Phi(\alpha_D)\}$, and component-wise to the diagonal of Λ , then $\exp(\Gamma)$ is a diagonal matrix with elements $\exp(\gamma_d)$, $d = 1, 2, \dots, D$ on its diagonal.

3.2. Preference learning system definition

Our goal is to design an efficient system for learning parameters of the latent user preference function. We have two integral components here: a user with his preferences and an agent or optimizer that interacts with the user by offering parameter settings. Therefore, we model the full preference learning system as a combination of two sub-systems, representing a user and an optimizer, see Fig. 1. In this system, the user can be seen as an environment in which the optimizer operates. Different settings for the user preference function, ϕ , correspond to different users or environments, and thus different user models. User responses to the trials, r , are observations of the environment. Moreover, they are reactions to the proposal trials. An optimizer or agent is a part of the preference learning system that interacts with the environment by generating trials or actions to which the environment reacts. The agent generates trials, according to some objective. The objective of our agent is to generate trials that result in accurate estimate of the user preferences for the parameters in as little number of trials as possible.

Definition 3.1. A preference learning system is defined by a combination of two interacting sub-systems $\{P^a, P^u\}$, an agent and a user, given by a sequence of $n = 1, 2, \dots$ distributions

$$P_n^a = p(\{\mathbf{x}_n^{\text{ref}}, \mathbf{x}_n^{\text{alt}}\} | \phi, r_1^{n-1}, \{\mathbf{x}_1^{\text{ref}}, \mathbf{x}_1^{\text{alt}}\}^{n-1}) \quad (15)$$

$$P_n^u = P(r_n | \phi, r_1^{n-1}, \{\mathbf{x}_1^{\text{ref}}, \mathbf{x}_1^{\text{alt}}\}^{n-1}, \{\mathbf{x}_n^{\text{ref}}, \mathbf{x}_n^{\text{alt}}\}), \quad (16)$$

where $p(\{\mathbf{x}_n^{\text{ref}}, \mathbf{x}_n^{\text{alt}}\} | \phi, r_1^{n-1}, \{\mathbf{x}_1^{\text{ref}}, \mathbf{x}_1^{\text{alt}}\}^{n-1})$ is a generative PDF for trials $\{\mathbf{x}_n^{\text{ref}}, \mathbf{x}_n^{\text{alt}}\}$ in the environment ϕ , given the previous sequence of observations, r_1^{n-1} , and trials, $\{\mathbf{x}_1^{\text{ref}}, \mathbf{x}_1^{\text{alt}}\}^{n-1}$; while $P(r_n | \phi, r_1^{n-1}, \{\mathbf{x}_1^{\text{ref}}, \mathbf{x}_1^{\text{alt}}\}^{n-1}, \{\mathbf{x}_n^{\text{ref}}, \mathbf{x}_n^{\text{alt}}\})$ is a generative probability of the response in this environment to the new trial $\{\mathbf{x}_n^{\text{ref}}, \mathbf{x}_n^{\text{alt}}\}$, given r_1^{n-1} and $\{\mathbf{x}_1^{\text{ref}}, \mathbf{x}_1^{\text{alt}}\}^{n-1}$.

Observe that, for the agent, described by (15), observations r_1^{n-1} are input data and actions $\{\mathbf{x}_n^{\text{ref}}, \mathbf{x}_n^{\text{alt}}\}$ are outputs, while for the user, described by (16), the situation is reversed.

In the preference learning system, defined as above, trials or actions are data that are produced by the agent, and every time a trial is presented, the user reacts to this *given trial*. As such the relation between trials and responses to trials becomes causal. Therefore, interactions between the user and the agent are characterized by a sequence of distributions

$$p(\phi), p(\{\mathbf{x}_1^{\text{ref}}, \mathbf{x}_1^{\text{alt}}\} | \phi), P(r_1 | \phi, \{\mathbf{x}_1^{\text{ref}}, \mathbf{x}_1^{\text{alt}}\}), p(\{\mathbf{x}_2^{\text{ref}}, \mathbf{x}_2^{\text{alt}}\} | \phi, r_1, \{\mathbf{x}_1^{\text{ref}}, \mathbf{x}_1^{\text{alt}}\}), \dots$$

3.3. Agent strategy design

Given the preference learning system, we have to deal with two connected problems. One problem relates to inference of the parameters of the user preference function. We look closely at this problem in Section 4. Another problem deals with efficient learning, and thus experimental design that leads to the limited but highly informative data that can be used to correctly perform inference. From this point of view, we need the agent to generate actions or trials that result in inferred user preference under which behavior of the environment is the same as its behavior under the true user preference ϕ . That is the same behavior over the history of taken actions and resulting observations. In what follows, we discuss and present our agent design.

3.3.1. Agent strategy

Observe that behavior of the environment is characterized by a sequence of distributions, given by (16), each indexed by the trials offered by the agent, since there is a causal relation between trials and user responses. In order to generate an informative action, the agent has to first evaluate the environment, based on its knowledge from the previous interactions. Thus, the agent has to assign a predictive distribution $Q(r_n | r_1^{n-1}, \{\mathbf{x}_1^{\text{ref}}, \mathbf{x}_1^{\text{alt}}\}^{n-1}, \{\mathbf{x}_n^{\text{ref}}, \mathbf{x}_n^{\text{alt}}\})$ for the next response, in order to assess the informativeness of the trial $\{\mathbf{x}_n^{\text{ref}}, \mathbf{x}_n^{\text{alt}}\}$. We start with defining the measure that is used to assess the performance of the agent, and then present the agent strategy that results from optimizing this measure.

The self-information loss function, also called log-loss, given by

$$-\log Q(r_n | r_1^{n-1}, \{\mathbf{x}_1^{\text{ref}}, \mathbf{x}_1^{\text{alt}}\}^{n-1}, \{\mathbf{x}_n^{\text{ref}}, \mathbf{x}_n^{\text{alt}}\}),$$

characterizes the quality of the Q -distribution assignment. The problem of probability assignment that minimizes self-information loss is equivalent to the problem of universal prediction under self-information loss, see [10]. In our case, though, prediction is not the main goal, but rather a mechanism to assess the validity of a statistical model and to further collect extra data (generate trial and observe the resulting user response) that can be used to improve the model. Under the expected self-information loss criterion, i.e.,

$$\mathbb{E}[-\log Q(r_n | r_1^{n-1}, \{\mathbf{x}_1^{\text{ref}}, \mathbf{x}_1^{\text{alt}}\}^{n-1}, \{\mathbf{x}_n^{\text{ref}}, \mathbf{x}_n^{\text{alt}}\})],$$

where expectation is taken with respect to the generative user response distribution (we also call it true user or true source distribution), the optimal predictive assignment is actually the assignment of the true conditional probability $P(r_n | \phi, r_1^{n-1}, \{\mathbf{x}_1^{\text{ref}}, \mathbf{x}_1^{\text{alt}}\}^{n-1}, \{\mathbf{x}_n^{\text{ref}}, \mathbf{x}_n^{\text{alt}}\})$. Here the true source model that governs generation of r_n belongs to the class of models, indexed by parameters ϕ , given by $\{P(\cdot | \phi, \cdot), \phi \in \Omega\}$. This true probability under certain ergodicity assumptions also minimizes the average self-information loss

$$\frac{1}{n} \sum_{l=1}^n -\log Q(r_l | r_1^{l-1}, \{\mathbf{x}_1^{\text{ref}}, \mathbf{x}_1^{\text{alt}}\}^{l-1}, \{\mathbf{x}_l^{\text{ref}}, \mathbf{x}_l^{\text{alt}}\}). \quad (17)$$

This sequential probability assignment gives rise to assignment of the probability Q to the entire observation sequence r_1^n and then sequential probability assignment and probability assignment problems for the whole sequence under self-information loss are equivalent. The average self-information loss for the true source distribution is, in fact, the entropy

$$H_n(P) = \mathbb{E}[-\log P(R_1^n | \phi, \{\mathbf{x}_1^{\text{ref}}, \mathbf{x}_1^{\text{alt}}\}^n)].$$

Observe that here we have conditioning on the trials, which are fixed agent's inputs. These trials $\{\mathbf{x}_1^{\text{ref}}, \mathbf{x}_1^{\text{alt}}\}^n$ can be modeled as side information, see e.g. [12], where side information is an additional information provided to the agent at the beginning of the next interaction. In our case, this side information is a causal function of previous interactions, see also (15).

In practice, the true source distribution is often unknown, and we have an extra loss, associated with using other than true distribution. This loss, which is coming on top of the minimum loss, is given by the information divergence between P and Q , also called the Kullback-Leibler (KL) divergence:

$$\begin{aligned} \mathbb{D}_n(P || Q) &\triangleq \mathbb{E}[-\log Q(R_1^n | \{\mathbf{x}_1^{\text{ref}}, \mathbf{x}_1^{\text{alt}}\}^n) - (-\log P(R_1^n | \phi, \{\mathbf{x}_1^{\text{ref}}, \mathbf{x}_1^{\text{alt}}\}^n))] \\ &= \mathbb{D}(P(R_1^n | \phi, \{\mathbf{x}_1^{\text{ref}}, \mathbf{x}_1^{\text{alt}}\}^n) || Q(R_1^n | \{\mathbf{x}_1^{\text{ref}}, \mathbf{x}_1^{\text{alt}}\}^n)), \end{aligned} \quad (18)$$

where expectation is taken with respect to P . Clearly, to be able to produce informative trials, the agent needs sufficient knowledge about the environment, and therefore should seek a probability assignment Q that minimizes this KL-divergence.

Recall that our agent has to perform universal prediction with respect to an indexed class of sources $\{P(\cdot|\phi, \cdot), \phi \in \Omega\}$. However, since ϕ is unknown, we take an approach, which has a strong Bayesian flavor, and assign some weighting distribution $w(\phi)$ on ϕ . Then the quality of distribution assignment Q will be assessed by the normalized weighted KL-divergence

$$D_n(Q, w, \{\mathbf{x}^{\text{ref}}, \mathbf{x}^{\text{alt}}\}) \triangleq \frac{1}{n} \int_{\Omega} w(\phi) \mathbb{D}_n(P||Q) d\phi. \quad (19)$$

Next, we present the agent strategy together with its characterization and then show why this is an efficient strategy.

Agent Strategy. The agent strategy for the sequential trial design in the preference learning system, defined by distributions $\{P^a, P^u\}$, is to

1. apply the following distributions as generative agent and predictive user distributions, respectively:

$$P_l^a = p_l = \delta(\mathbf{x}_l^{\text{ref}} - \mathbf{x}_{l-1}^{\text{alt}})^{r_{l-1}} \cdot \delta(\mathbf{x}_l^{\text{ref}} - \mathbf{x}_{l-1}^{\text{ref}})^{1-r_{l-1}} \cdot \int p(\mathbf{x}_l^{\text{alt}}, \gamma | r_1^{l-1}, \{\mathbf{x}^{\text{ref}}, \mathbf{x}^{\text{alt}}\}_1^{l-1}) d\gamma$$

$$P_l^u = Q_{p,l} = \int_{\Omega} p(\phi | r_1^{l-1}, \{\mathbf{x}^{\text{ref}}, \mathbf{x}^{\text{alt}}\}_1^{l-1}) \cdot P(r_l | \phi, \{\mathbf{x}^{\text{ref}}, \mathbf{x}^{\text{alt}}\}_1^{l-1}, \{\mathbf{x}_l^{\text{ref}}, \mathbf{x}_l^{\text{alt}}\}) d\phi,$$

for $l = 1, 2, \dots, n$, where

$$p(\phi | r_1^{l-1}, \{\mathbf{x}^{\text{ref}}, \mathbf{x}^{\text{alt}}\}_1^{l-1}) = \frac{p(\phi) p(r_1^{l-1} | \phi, \{\mathbf{x}^{\text{ref}}, \mathbf{x}^{\text{alt}}\}_1^{l-1})}{\int_{\Omega} p(\phi') p(r_1^{l-1} | \phi', \{\mathbf{x}^{\text{ref}}, \mathbf{x}^{\text{alt}}\}_1^{l-1}) d\phi'}$$

is a posterior distribution and $\int p(\mathbf{x}_l^{\text{alt}}, \gamma | r_1^{l-1}, \{\mathbf{x}^{\text{ref}}, \mathbf{x}^{\text{alt}}\}_1^{l-1}) d\gamma$ is a marginal of the location parameter of this posterior;

2. generate the corresponding sequence of trials, for $l = 2, 3, \dots, n$, based on the history of interactions with the user, according to the following rule

$$\{\mathbf{x}_l^{\text{ref}}, \mathbf{x}_l^{\text{alt}}\} = \arg \max_{\{\mathbf{x}^{\text{ref}}, \mathbf{x}^{\text{alt}}\} : P_l^a} I(R; \phi | R_1^{l-1}, \{\mathbf{x}^{\text{ref}}, \mathbf{x}^{\text{alt}}\}_1^{l-1}, \{\mathbf{x}^{\text{ref}}, \mathbf{x}^{\text{alt}}\}),$$

where $I(\cdot; \cdot)$ is mutual information.

This strategy is asymptotically optimal in that it minimizes the normalized weighted KL-divergence. Given this strategy, for n interactions, the achieved normalized weighted KL-divergence is given by

$$D_n(P^u, \{\mathbf{x}^{\text{ref}}, \mathbf{x}^{\text{alt}}\}) = \frac{1}{n} \sum_{l=1}^n I(R_l; \phi | R_1^{l-1}, \{\mathbf{x}^{\text{ref}}, \mathbf{x}^{\text{alt}}\}_1^{l-1}, \{\mathbf{x}_l^{\text{ref}}, \mathbf{x}_l^{\text{alt}}\}).$$

3.3.2. Predictive user response distribution

In what follows, we discuss the principles guiding the agent strategy. In order to assign a good predictive probability, the agent has to find w and Q such that (19) is minimized. One way to tackle the problem is to apply a maximin criterion that results in a mixture approach with a Bayesian flavor, i.e.,

$$\sup_w \inf_Q D_n(Q, w, \{\mathbf{x}^{\text{ref}}, \mathbf{x}^{\text{alt}}\}). \quad (20)$$

For a given $w(\phi)$, the minimizing distribution is given by the weighted distribution

$$Q_w(r_1^n | \{\mathbf{x}^{\text{ref}}, \mathbf{x}^{\text{alt}}\}_1^n) = \int_{\Omega} w(\phi) P(r_1^n | \phi, \{\mathbf{x}^{\text{ref}}, \mathbf{x}^{\text{alt}}\}_1^n) d\phi, \quad (21)$$

and the corresponding weighted divergence is given by $I_w(R_1^n; \phi | \{\mathbf{x}^{\text{ref}}, \mathbf{x}^{\text{alt}}\}_1^n)$, where subscript w indicates that some weighting distribution $w(\phi)$ is used. The agent, however, has to assign the predictive probabilities sequentially, after every interaction. Sequential assignment gives rise to the sequence probability assignment. It was shown in [13] that the above mixture of

$$\{P(r_1^n | \phi, \{\mathbf{x}^{\text{ref}}, \mathbf{x}^{\text{alt}}\}_1^n), \phi \in \Omega\}$$

can be represented by the mixture of conditional probability functions

$$\{P(r_n | \phi, r_1^{n-1}, \{\mathbf{x}^{\text{ref}}, \mathbf{x}^{\text{alt}}\}_1^{n-1}, \{\mathbf{x}_n^{\text{ref}}, \mathbf{x}_n^{\text{alt}}\}), \phi \in \Omega\},$$

as

$$Q_w(r_n | r_1^{n-1}, \{\mathbf{x}^{\text{ref}}, \mathbf{x}^{\text{alt}}\}_1^{n-1}, \{\mathbf{x}_n^{\text{ref}}, \mathbf{x}_n^{\text{alt}}\}) = \int_{\Omega} w(\phi | r_1^{n-1}, \{\mathbf{x}^{\text{ref}}, \mathbf{x}^{\text{alt}}\}_1^{n-1}) \cdot P(r_n | \phi, \{\mathbf{x}^{\text{ref}}, \mathbf{x}^{\text{alt}}\}_1^{n-1}, \{\mathbf{x}_n^{\text{ref}}, \mathbf{x}_n^{\text{alt}}\}) d\phi, \quad (22)$$

where the weighting function is a posterior distribution of ϕ given r_1^{n-1} and $\{\mathbf{x}^{\text{ref}}, \mathbf{x}^{\text{alt}}\}_1^{n-1}$, i.e.,

$$\begin{aligned} w(\phi | r_1^{n-1}, \{\mathbf{x}^{\text{ref}}, \mathbf{x}^{\text{alt}}\}_1^{n-1}) &= p(\phi | r_1^{n-1}, \{\mathbf{x}^{\text{ref}}, \mathbf{x}^{\text{alt}}\}_1^{n-1}) \\ &= \frac{p(\phi) p(r_1^{n-1} | \phi, \{\mathbf{x}^{\text{ref}}, \mathbf{x}^{\text{alt}}\}_1^{n-1})}{\int_{\Omega} p(\phi') p(r_1^{n-1} | \phi', \{\mathbf{x}^{\text{ref}}, \mathbf{x}^{\text{alt}}\}_1^{n-1}) d\phi'}, \end{aligned} \quad (23)$$

where the prior on the user preference parameters is independent of trials. Moreover, the performance of this sequential assignment is the same as the performance of the assignment when the whole sequence is observed. The minimizing distribution functions Q_w , in their turn, correspond to the normalized weighted information divergence given by

$$\begin{aligned} D_n(Q_w, \{\mathbf{x}^{\text{ref}}, \mathbf{x}^{\text{alt}}\}) &= \frac{1}{n} \sum_{l=1}^n \sum_{r_l \in \{0,1\}} \int_{\Omega} p(\phi | r_1^{l-1}, \{\mathbf{x}^{\text{ref}}, \mathbf{x}^{\text{alt}}\}_1^{l-1}) \cdot P(r_l | \phi, r_1^{l-1}, \{\mathbf{x}^{\text{ref}}, \mathbf{x}^{\text{alt}}\}_1^{l-1}, \{\mathbf{x}_l^{\text{ref}}, \mathbf{x}_l^{\text{alt}}\}) \\ &\quad \cdot \ln \frac{P(r_l | \phi, r_1^{l-1}, \{\mathbf{x}^{\text{ref}}, \mathbf{x}^{\text{alt}}\}_1^{l-1}, \{\mathbf{x}_l^{\text{ref}}, \mathbf{x}_l^{\text{alt}}\})}{P(r_l | \{\mathbf{x}^{\text{ref}}, \mathbf{x}^{\text{alt}}\}_1^{l-1}, \{\mathbf{x}_l^{\text{ref}}, \mathbf{x}_l^{\text{alt}}\})} d\phi \\ &= \frac{1}{n} \sum_{l=1}^n I(R_l; \phi | R_1^{l-1}, \{\mathbf{x}^{\text{ref}}, \mathbf{x}^{\text{alt}}\}_1^{l-1}, \{\mathbf{x}_l^{\text{ref}}, \mathbf{x}_l^{\text{alt}}\}). \end{aligned} \quad (24)$$

Here we have made a choice for the weighting functions up to the prior $p(\phi)$, given by (23). This is a sequential form of writing out $D_n(Q_w, \{\mathbf{x}^{\text{ref}}, \mathbf{x}^{\text{alt}}\})$ that coincides with the non-sequential result given by $I_w(R^n; \phi | \{\mathbf{x}^{\text{ref}}, \mathbf{x}^{\text{alt}}\}_1^n)$.

Note that in this universal prediction problem, the goal is to find a probability assignment that guarantees that the predictor performs well for all ϕ . The supremum of $1/n I_w(R^n; \phi | \{\mathbf{x}^{\text{ref}}, \mathbf{x}^{\text{alt}}\}_1^n)$ over all allowable $w(\phi)$ can be interpreted as a capacity of the “channel” between the channel output R_1^n and the channel input ϕ with side information $\{\mathbf{x}^{\text{ref}}, \mathbf{x}^{\text{alt}}\}_1^n$, defined by the class of sources $\{P(\cdot | \phi, \cdot), \phi \in \Omega\}$. This capacity is given by $C_n \stackrel{\Delta}{=} 1/n I_{w^*}(R^n; \phi | \{\mathbf{x}^{\text{ref}}, \mathbf{x}^{\text{alt}}\}_1^n)$, where $w^*(\cdot)$ is the capacity achieving prior, and then it holds that $\mathbb{D}_n(P | Q_w) \leq n C_n$, $\forall \phi$, guarantees in the minimax sense that

$$\mathbb{D}_n(P | Q) \geq n C_n. \quad (25)$$

On the other hand, the strong converse theorem of [14] states that under self-information loss, for any asymptotically good approximation of $w^*(\phi)$, there is a remarkable concentration phenomenon, stating that, for any $\varepsilon > 0$ and w^* -most values of ϕ , it holds that

$$\mathbb{D}_n(P | Q) \geq (1 - \varepsilon) n C_n, \quad \forall Q. \quad (26)$$

The practical problem is then finding the capacity achieving prior. However, the result in [15] allows $w(\cdot)$ to be an arbitrary weight function, thus telling that Q_w is optimal for w -most points in Ω :

$$\mathbb{D}_n(P | Q) \geq \mathbb{D}_n(P | Q_w). \quad (27)$$

Let's look at the interpretation of this result. This capacity gives information on the richness of the source class. The capacity-achieving weighting function assigns higher weights to the regions, where the sources are better discriminated and data is more informative about the parameters; and lower weights to the regions, where the sources are closer. Therefore, in the problem of preference learning, the use of posterior distribution of ϕ , as weighting functions, helps us to evaluate usefulness of the data generated by the agent for inference of ϕ . We need to distinguish a single unknown source. Hence, on one hand, we aim at attaining the lower bound on the information divergence and, on the other hand, at finding good $w(\cdot)$ that further reduces this bound. In Section 4, we further discuss approximation of the weighting distributions in their sequential form.

3.3.3. Generative agent distribution

Now that we have defined the predictor Q_w to be used by the agent for preference learning system characterization, we need to specify the generative agent distribution. From the optimization point of view, the agent has to generate the proposals that lead to the optimal user preferences as quickly as possible. This means that, for every interaction with a user, the agent has to provide a better alternative than a reference proposal, with high probability. This can be seen as a sort of gradient ascent method. Clearly, this alternative would correspond to the user preference or the agent's estimate of it. Therefore, the generative distribution of the alternative proposal generation would coincide with the distribution of the optimal parameters corresponding to the user preference. This is a marginal posterior distribution of location parameters.

Given the fact that the next reference proposal is just a deterministic function of the previous trial and user response to it, the generative process for the next trial l is given by

$$p(\{\mathbf{x}_l^{\text{ref}}, \mathbf{x}_l^{\text{alt}}\} | \phi, r_1^{l-1}, \{\mathbf{x}^{\text{ref}}, \mathbf{x}^{\text{alt}}\}_1^{l-1}) \stackrel{\Delta}{=} \delta(\mathbf{x}_l^{\text{ref}} - \mathbf{x}_{l-1}^{\text{alt}})^{r_{l-1}} \cdot \delta(\mathbf{x}_l^{\text{ref}} - \mathbf{x}_{l-1}^{\text{ref}})^{1-r_{l-1}} \cdot \int p(\mathbf{x}_l^{\text{alt}}, \gamma | r_1^{l-1}, \{\mathbf{x}^{\text{ref}}, \mathbf{x}^{\text{alt}}\}_1^{l-1}) d\gamma, \quad (28)$$

where $p(\mathbf{x}_l^{\text{alt}}, \gamma | r_1^{l-1}, \{\mathbf{x}_1^{\text{ref}}, \mathbf{x}_1^{\text{alt}}\}_{l-1}^l)$ coincides with an expanded version of posterior (23). Thus, the agent generative distribution functions p_l^a are given by (28), for $l = 2, 3, \dots, n$.

3.3.4. Trial design

Next, we look at the problem of trial generation from a data informativeness point of view. The agent trial generation process is given by (28). The trials, generated by the agent, result in user responses, characterized by the predictor Q_w , which performance is given by the normalized weighted information divergence (24). As discussed above, this information divergence tells us how well data helps to distinguish the source, i.e., the individual user preference ϕ . As such, we can also interpret this performance measure as the agent uncertainty about the user that we will call remaining system uncertainty (RSU).

For the normalized weighted information divergence, trials serve as a causal side information and are given values produced by the agent. The generative agent distribution provides the “direction” of the data being sought for. However, to facilitate efficient learning of the optimal parameters, we also need to produce the most informative trials. To evaluate trial informativeness, we use the normalized weighted information divergence, by looking at it as a function of trials.

Since trials cause user responses, we gain the most information from getting a user response on the trial, about which our predictor is most uncertain. Thus, to generate the next trial at time n , the agent should search for the worst case behavior of the optimal predictor from the trial (data) perspective, i.e.,

$$\begin{aligned} \{\mathbf{x}_n^{\text{ref}}, \mathbf{x}_n^{\text{alt}}\} &= \arg \max_{\{\mathbf{x}_n^{\text{ref}}, \mathbf{x}_n^{\text{alt}}\} : p_n^a} D_n(Q_w, \{\mathbf{x}_n^{\text{ref}}, \mathbf{x}_n^{\text{alt}}\}) \\ &= \arg \max_{\{\mathbf{x}_n^{\text{ref}}, \mathbf{x}_n^{\text{alt}}\} : p_n^a} \sum_{l=1}^n I(R_l; \phi | R_1^{l-1}, \{\mathbf{x}_1^{\text{ref}}, \mathbf{x}_1^{\text{alt}}\}_{l-1}^l, \{\mathbf{x}_l^{\text{ref}}, \mathbf{x}_l^{\text{alt}}\}) \\ &= \arg \max_{\{\mathbf{x}_n^{\text{ref}}, \mathbf{x}_n^{\text{alt}}\} : p_n^a} I(R_n; \phi | R_1^{n-1}, \{\mathbf{x}_1^{\text{ref}}, \mathbf{x}_1^{\text{alt}}\}_{n-1}^n, \{\mathbf{x}_n^{\text{ref}}, \mathbf{x}_n^{\text{alt}}\}), \end{aligned} \quad (29)$$

here due to sequential nature of interactions between the agent and the user, the next trial to be generated by the agent corresponds to the argument of the maximum of the mutual information between the predicted response for a given next trial and optimal parameters, given the history of interactions. We call this last term under arg max weighted individual divergence. This concludes the agent strategy for the trial design.

4. Bayesian inference for the user model

In the previous section, we defined the preference learning system and the experimental design strategy to obtain the data required for inference of the user model or, equivalently, parameters of the user preference function. The latter is, in fact, our main goal. Indeed, user model inference and experimental design are intimately related. In the experimental design, we are also concerned with user model inference, but there it is used as a weighting distribution. In general, user and agent interactions are sequential, and, in the experimental design, the agent also assigns predictive probability sequentially. Therefore, we concentrate and present a sequential Bayesian procedure for user model inference.

Recall that in the predictive probability assignment procedure, we have specified the weighting (posterior) distribution of the user model parameters up to the prior $p(\phi)$. Therefore, we start here with this prior specification and assume the following prior distribution for the user preference parameters:

$$p(\phi) = \mathcal{N}(\phi | \mu_\phi, \Sigma_\phi). \quad (30)$$

This assumption is the consequence of assumption that the user has one optimal preference, which means that preference distribution should have a clear optimum. Next, consider data points from the user-agent interactions in the preference learning system, denoted by $D^{n-1} = \{r_l, \{\mathbf{x}_l^{\text{ref}}, \mathbf{x}_l^{\text{alt}}\}\}_{l=1}^{n-1}$. Given these data points, we work out the posterior (23) as

$$\begin{aligned} p(\phi | r_1^{n-1}, \{\mathbf{x}_1^{\text{ref}}, \mathbf{x}_1^{\text{alt}}\}_{1}^{n-1}) &= \frac{p(\phi)p(r_1^{n-1} | \phi, \{\mathbf{x}_1^{\text{ref}}, \mathbf{x}_1^{\text{alt}}\}_{1}^{n-1})}{\int p(\phi')p(r_1^{n-1} | \phi', \{\mathbf{x}_1^{\text{ref}}, \mathbf{x}_1^{\text{alt}}\}_{1}^{n-1})d\phi'} \\ &= p(\phi) \cdot \prod_{l=1}^{n-1} \frac{p(r_l | \phi, r_1^{l-1}, \{\mathbf{x}_1^{\text{ref}}, \mathbf{x}_1^{\text{alt}}\}_{l-1}^l, \{\mathbf{x}_l^{\text{ref}}, \mathbf{x}_l^{\text{alt}}\})}{P(D^l)} \\ &= p(\phi) \cdot \prod_{l=1}^{n-1} \frac{p(r_l | \phi, \{\mathbf{x}_l^{\text{ref}}, \mathbf{x}_l^{\text{alt}}\})}{P(D^l)} \\ &= \mathcal{N}(\phi | \mu_\phi, \Sigma_\phi) \cdot \prod_{l=1}^{n-1} \frac{1}{C_l} \cdot \Phi(f(\mathbf{x}_l^{\text{ref}}; \phi) - f(\mathbf{x}_l^{\text{alt}}; \phi))^{r_l} \cdot \Phi(f(\mathbf{x}_l^{\text{alt}}; \phi) - f(\mathbf{x}_l^{\text{ref}}; \phi))^{1-r_l}, \end{aligned} \quad (31)$$

where C_l are normalizing constants, and where, in the third equality, we used the Markov property, since the user response only depends on the current trial, given parameters of the preference function. While this expression provides us with an analytical expression for the posterior, its exact estimation and use in the trial design is problematic, due to its non-linear form and complex integral evaluations. Therefore, to make the problem tractable, we turn to approximate sequential estimation, based on Assumed Density Filtering (ADF), also called moment matching, see, e.g., [16] and [17]. The idea is to use a Gaussian distribution as an approximating distribution for the posterior. Note, that our assumption that a user has a single optimal preference means that the posterior should manifest it by having a peak around true preference, which, in its turn, makes Gaussian approximation to be a viable assumption. The ADF approach allows for a simple update of the first and second moments of the posterior after each trial.

Following the ADF approach, we re-write the posterior distribution in the following way, by defining $\tilde{p}_0(\boldsymbol{\phi}) \triangleq p(\boldsymbol{\phi})$, $\tilde{p}_l(\boldsymbol{\phi}) \triangleq p(r_l | \boldsymbol{\phi}, \{\mathbf{x}_l^{\text{ref}}, \mathbf{x}_l^{\text{alt}}\})$,

$$p(\boldsymbol{\phi} | D^{n-1}) = \prod_{l=1}^{n-1} \frac{1}{C_l} \cdot \tilde{p}_l(\boldsymbol{\phi}) \approx q_{n-1}(\boldsymbol{\phi}), \quad (32)$$

where $q_{n-1}(\boldsymbol{\phi})$ is our approximating family, which we choose to be a Gaussian one, i.e.,

$$q_{n-1}(\boldsymbol{\phi}) = \mathcal{N}(\boldsymbol{\phi} | \boldsymbol{\mu}_{\phi, n-1}, \Sigma_{\phi, n-1}). \quad (33)$$

Observe that initially, at the beginning of interaction, we only have the prior parameter distribution and do not need any approximation. Then, after each trial, we need to incorporate new evidence and perform an approximation. In order to do this, after each trial l , we take the “exact” posterior

$$\hat{p}(\boldsymbol{\phi} | D^l) = \frac{\tilde{p}_l(\boldsymbol{\phi}) q_{l-1}(\boldsymbol{\phi})}{\int \tilde{p}_l(\boldsymbol{\phi}) q_{l-1}(\boldsymbol{\phi}) d\boldsymbol{\phi}} \quad (34)$$

and find the approximating distribution $q_l(\cdot)$ that minimizes the KL-divergence:

$$q_l(\boldsymbol{\phi}) : \min_{q(\cdot)} \mathbb{D}(\hat{p}(\boldsymbol{\phi} | D^l) || q(\boldsymbol{\phi})), \quad (35)$$

given that $q_l(\cdot)$ belongs to the Gaussian approximation family. This problem is equivalent to the maximum likelihood estimation problem for $\hat{p}(\boldsymbol{\phi} | D^l)$, and then the ADF procedure reduces to propagating the first two moments:

$$\boldsymbol{\mu}_{\phi, l} = \mathbb{E}_{q_l(\boldsymbol{\phi})}[\boldsymbol{\phi}] = \mathbb{E}_{\hat{p}(\boldsymbol{\phi} | D^l)}[\boldsymbol{\phi}], \quad (36)$$

$$\begin{aligned} \Sigma_{\phi, l} &= \mathbb{E}_{q_l(\boldsymbol{\phi})}[(\boldsymbol{\phi} - \boldsymbol{\mu}_{\phi})^T (\boldsymbol{\phi} - \boldsymbol{\mu}_{\phi})] \\ &= \mathbb{E}_{\hat{p}(\boldsymbol{\phi} | D^l)}[(\boldsymbol{\phi} - \boldsymbol{\mu}_{\phi})^T (\boldsymbol{\phi} - \boldsymbol{\mu}_{\phi})]. \end{aligned} \quad (37)$$

Thus, the problem of user model inference reduces to calculations of the moments of the “true” posteriors. Observe that the first moment, given by (36)-(37), provides us with the (current) estimate of the parameters of the user preference function.

5. Implementation of the sequential procedure for active user model inference

After specification of the experimental design and user model inference approach, we describe here implementation details for sequential active user model inference. We start with initialization of the experimental design. To perform an initial trial, first two proposals to be evaluated are selected uniformly at random, i.e., \mathbf{x} and \mathbf{x}' according to Uniform $([0, 1]^D)$. Based on the user evaluation, they form the first trial $\{\mathbf{x}_1^{\text{ref}}, \mathbf{x}_1^{\text{alt}}\}$. Furthermore, the model prior is given by (30) as

$$p(\boldsymbol{\phi}) = \mathcal{N}(\boldsymbol{\phi} | \boldsymbol{\mu}_{\phi, 0}, \Sigma_{\phi, 0}),$$

where $\boldsymbol{\mu}_{\phi, 0}, \Sigma_{\phi, 0}$ are hyperparameters.

After each interaction l , $l = 1, 2, \dots, n$, we obtain the user feedback on the trial, and thus the next data point $\{r_l, \{\mathbf{x}_l^{\text{ref}}, \mathbf{x}_l^{\text{alt}}\}\}$. At each step, we need to calculate the moments of the posterior distribution $\boldsymbol{\mu}_{\phi, l}, \Sigma_{\phi, l}$. To avoid calculations of integrals for non-linear distribution functions, we use Monte-Carlo simulations to infer the user model distribution. We apply the Metropolis–Hastings (MH) sampling algorithm, for D_l , according to the following model:

$$\boldsymbol{\phi} \sim \mathcal{N}(\boldsymbol{\mu}_{\phi, l-1}, \Sigma_{\phi, l-1}) \quad (38)$$

$$r_l \sim \text{Bernoulli}(\Phi(f(\mathbf{x}_l^{\text{alt}}; \boldsymbol{\phi}) - f(\mathbf{x}_l^{\text{ref}}; \boldsymbol{\phi}))). \quad (39)$$

The MH simulations result in a sequence of samples or posterior sample, denoted by $\tilde{q}_l(\boldsymbol{\phi})$. This sample is further used to calculate the corresponding moments, using maximum likelihood estimation:

$$\boldsymbol{\mu}_{\phi, l} = \frac{1}{M} \sum_{\boldsymbol{\phi}_j \sim \tilde{q}_l(\boldsymbol{\phi})} \boldsymbol{\phi}_j, \quad (40)$$

$$\Sigma_{\phi,l} = \frac{1}{M-1} \sum_{\phi_j \sim q_l(\phi)} (\phi_j - \mu_{\phi,l})^T (\phi_j - \mu_{\phi,l}). \quad (41)$$

Hence, we obtain approximate posterior, given by $q_l(\phi) = \mathcal{N}(\phi | \mu_{\phi,l}, \Sigma_{\phi,l})$, for $l = 1, 2, \dots, n$.

To perform the trial design at step l , we use the approximate posterior as a weighting distribution. Then, we need to calculate RSU in step l , given by the mutual information in (29). This mutual information expression can be further worked out as

$$I(R; \phi | R_1^{l-1}, \{\mathbf{x}^{\text{ref}}, \mathbf{x}^{\text{alt}}\}_1^{l-1}, \{\mathbf{x}^{\text{ref}}, \mathbf{x}^{\text{alt}}\}) \approx H(R | D^{l-1}, \{\mathbf{x}^{\text{ref}}, \mathbf{x}^{\text{alt}}\}) - \mathbb{E}_{q_l(\phi)} [H(R | \phi, \{\mathbf{x}^{\text{ref}}, \mathbf{x}^{\text{alt}}\})], \quad (42)$$

where approximation is the result of using approximate posterior instead of the exact one. Recall that our main goal is, in fact, inference of the location parameter in the user model. Therefore, we decided to treat the sensitivity parameter as a nuisance parameter. In this case, we have to modify the objective, given above, by integrating out the sensitivity parameter and, hence, arrive at the following expression:

$$I(R; \alpha | R_1^{l-1}, \{\mathbf{x}^{\text{ref}}, \mathbf{x}^{\text{alt}}\}_1^{l-1}, \{\mathbf{x}^{\text{ref}}, \mathbf{x}^{\text{alt}}\}) \approx H(R | D^{l-1}, \{\mathbf{x}^{\text{ref}}, \mathbf{x}^{\text{alt}}\}) - \mathbb{E}_{q_l(\alpha)} \left[H \left(\mathbb{E}_{q_l(\gamma | \alpha)} [R | \alpha, \gamma, \{\mathbf{x}^{\text{ref}}, \mathbf{x}^{\text{alt}}\}] \right) \right]. \quad (43)$$

Then, since our joint approximate posterior is Gaussian, which first and second moments can be written as

$$\mu_{\phi,l} = \begin{bmatrix} \mu_{\alpha,l} \\ \mu_{\gamma,l} \end{bmatrix}, \quad \Sigma_{\phi,l} = \begin{bmatrix} \Sigma_{\alpha,l} & \Sigma_{\alpha\gamma,l} \\ \Sigma_{\gamma\alpha,l} & \Sigma_{\gamma,l} \end{bmatrix}, \quad (44)$$

the approximate marginal $q_l(\alpha)$ is also Gaussian

$$q_l(\alpha) = \mathcal{N}(\alpha | \mu_{\alpha,l}, \Sigma_{\alpha,l}), \quad (45)$$

and so the conditional distributions $q_l(\gamma | \alpha)$, which expectation and covariance are given, according to the Markov-Gauss theorem, by

$$\mu_{\gamma | \alpha,l} = \mu_{\gamma,l} + \Sigma_{\gamma\alpha,l} \Sigma_{\alpha,l}^{-1} (\alpha - \mu_{\alpha,l}) \quad (46)$$

$$\Sigma_{\gamma | \alpha,l} = \Sigma_{\gamma,l} - \Sigma_{\gamma\alpha,l} \Sigma_{\alpha,l}^{-1} \Sigma_{\alpha\gamma,l}. \quad (47)$$

Observe that decomposition of (43) allows for easier computation of the mutual information, since the entropy terms are binary entropy functions, defined by $h(p) \triangleq -p \log_2 p - (1-p) \log_2 (1-p)$. Furthermore, we approximate expectations by averages and use approximate posterior to calculate the entropy terms in (43) as follows

$$\begin{aligned} H(R | D^{l-1}, \{\mathbf{x}^{\text{ref}}, \mathbf{x}^{\text{alt}}\}) &= h(p(r | D^l, \{\mathbf{x}^{\text{ref}}, \mathbf{x}^{\text{alt}}\})) \\ &= h \left(\int p(r | \phi, \{\mathbf{x}^{\text{ref}}, \mathbf{x}^{\text{alt}}\}) \cdot p(\phi | D^{l-1}) d\phi \right) \\ &\approx h \left(\frac{1}{M} \cdot \sum_{\{\phi_m\}_{m=1}^M \sim q_l(\phi)} \Phi(v(\phi_m)) \right) \\ &= h \left(\frac{1}{M} \cdot \sum_{\{\phi_m\}_{m=1}^M \sim \mathcal{N}(\phi_m | \mu_{\phi,l}, \Sigma_{\phi,l})} \Phi(v(\phi_m)) \right), \end{aligned} \quad (48)$$

where $v(\phi_m) \triangleq f(\mathbf{x}^{\text{ref}}; \phi_m) - f(\mathbf{x}^{\text{alt}}; \phi_m)$;

$$\begin{aligned} &\mathbb{E}_{q_l(\alpha)} \left[H \left(\mathbb{E}_{q_l(\gamma | \alpha)} [R | \alpha, \gamma, \{\mathbf{x}^{\text{ref}}, \mathbf{x}^{\text{alt}}\}] \right) \right] \\ &\approx \frac{1}{M} \sum_{\{\alpha_m\}_{m=1}^M \sim q_l(\alpha)} h \left(\frac{1}{M'} \sum_{\{\gamma_{m'}\}_{m'=1}^{M'} \sim q_l(\gamma | \alpha_m)} \Phi(v(\alpha_m, \gamma_{m'})) \right) \\ &= \frac{1}{M} \sum_{\{\alpha_m\}_{m=1}^M \sim \mathcal{N}(\alpha_m | \mu_{\alpha,l}, \Sigma_{\alpha,l})} h \left(\frac{1}{M'} \sum_{\{\gamma_{m'}\}_{m'=1}^{M'} \sim \mathcal{N}(\gamma_{m'} | \mu_{\gamma | \alpha,l}, \Sigma_{\gamma | \alpha,l})} \Phi(v(\alpha_m, \gamma_{m'})) \right), \end{aligned} \quad (49)$$

where $v(\alpha_m, \gamma_{m'})$ is defined in a similar way as above, but with expanded parameters, see also (14).

Next, the generative agent distribution for the trials is given by

$$p_l^a = \delta(\mathbf{x}_l^{\text{ref}} - \mathbf{x}_{l-1}^{\text{alt}})^{r_{l-1}} \cdot \delta(\mathbf{x}_l^{\text{ref}} - \mathbf{x}_{l-1}^{\text{ref}})^{1-r_{l-1}} \cdot \mathcal{N}(\mathbf{x}_l^{\text{alt}} | \mu_{\alpha,l}, \Sigma_{\alpha,l}). \quad (50)$$

Therefore, generating trials from this distribution is equivalent to setting

Algorithm 1: Trial design.**Data:**

- $q_{l-1}(\phi)$ – approximate posterior after $l-1$ interactions
- $\{\mathbf{x}_{l-1}^{\text{ref}}, \mathbf{x}_{l-1}^{\text{alt}}\}$ – trial at $l-1$ -th interaction
- r_{l-1} – reply to $l-1$ -th trial
- M, M' – number of Monte-Carlo samples
- N – number of samples for alternative proposal generation

Result:

- $\{\mathbf{x}_l^{\text{ref}}, \mathbf{x}_l^{\text{alt}}\}$ – new trial
- D_l – system uncertainty or weighted individual KL-divergence of l -th step

```

1  $\mathbf{x}_l^{\text{ref}} \leftarrow (\mathbf{x}_{l-1}^{\text{alt}})^{r_{l-1}} \cdot (\mathbf{x}_{l-1}^{\text{ref}})^{1-r_{l-1}};$ 
2  $\mathbf{x}_l^{\text{alt}} \leftarrow \text{Sample} (q_{l-1}(\phi));$ 
3  $D_l \leftarrow \text{MI} (M, M', q_{l-1}(\phi), \mathbf{x}_l^{\text{alt}}, \mathbf{x}_l^{\text{ref}});$ 
4 for  $i \leftarrow 1$  to  $N-1$  do
5    $\mathbf{x}_l^{\text{alt}} \leftarrow \text{Sample} (q_{l-1}(\phi));$ 
6    $D \leftarrow \text{MI} (M, M', q_{l-1}(\phi), \mathbf{x}_l^{\text{alt}}, \mathbf{x}_l^{\text{ref}});$ 
7   if  $D_i < D$  then
8      $\mathbf{x}_l^{\text{alt}} \leftarrow \mathbf{x}_i^{\text{alt}};$ 
9      $D_i \leftarrow D$ 
10  end
11 end

```

Algorithm 2: Learning procedure.**Data:**

- $q_0(\phi)$ – model prior
- $\{\mathbf{x}_1^{\text{ref}}, \mathbf{x}_1^{\text{alt}}\}$ – initial trial
- r_1 – reply to the initial trial
- M, M' – number of Monte-Carlo samples

Result:

- $q(\phi)$ – user model parameter distribution
- $\mathbb{E}_q(\alpha)$ – optimal parameter estimate
- D – remaining system uncertainty

```

1  $l \leftarrow 0;$ 
2  $D_l \leftarrow \infty;$ 
3  $D \leftarrow 0;$ 
4 while  $D_l > l\delta$  do
5    $l \leftarrow l+1;$ 
6    $\{\mathbf{x}_l^{\text{ref}}, \mathbf{x}_l^{\text{alt}}\}, D_l \leftarrow \text{TrialDesign} (q_{l-1}(\phi), \{\mathbf{x}_{l-1}^{\text{ref}}, \mathbf{x}_{l-1}^{\text{alt}}\}, r_{l-1});$ 
7    $D \leftarrow D + D_l;$ 
8    $r_l \leftarrow \text{Trial} (\{\mathbf{x}_l^{\text{ref}}, \mathbf{x}_l^{\text{alt}}\});$ 
9    $q_l(\phi) \leftarrow \text{MC} (M, \mu_{\phi,l-1}, \Sigma_{\phi,l-1}, \{\mathbf{x}_l^{\text{ref}}, \mathbf{x}_l^{\text{alt}}\}, r_l)$ 
10 end
11  $q(\phi) \leftarrow q_l(\phi);$ 
12  $\mathbb{E}_q(\alpha) \leftarrow \mu_{\alpha,l};$ 
13  $D \leftarrow D/l;$ 

```

$$\mathbf{x}_l^{\text{ref}} = (\mathbf{x}_{l-1}^{\text{alt}})^{r_{l-1}} \cdot (\mathbf{x}_{l-1}^{\text{ref}})^{1-r_{l-1}} \quad (51)$$

and generating an alternative proposal according to

$$\mathbf{x}_l^{\text{alt}} \sim \mathcal{N}(\mathbf{x}_l^{\text{alt}} | \mu_{\alpha,l}, \Sigma_{\alpha,l}). \quad (52)$$

Finally, given the expressions above, the trial design constitutes generating a pool of alternative proposals, using (52), calculating mutual information terms for the resulting trials, using (48) and (49), and selecting the alternative that corresponds to the highest mutual information term. This procedure is also given in Algorithm 1, where $\text{MI}(\cdot)$ is computed, using (48) and (49).

Our learning procedure can be summarized as a sequential process of trials and user model inferences, with the stopping criterion, determined by convergence of the normalized weighted KL-divergence, in a sense that convergence is achieved, when uncertainty reduction becomes smaller than some small predefined δ . The learning procedure is also provided as a pseudo-code in Algorithm 2. There $\text{MC}(\cdot)$ denotes Monte-Carlo simulations according to the model, given by (38)-(39), followed by calculations in (40) and (41).

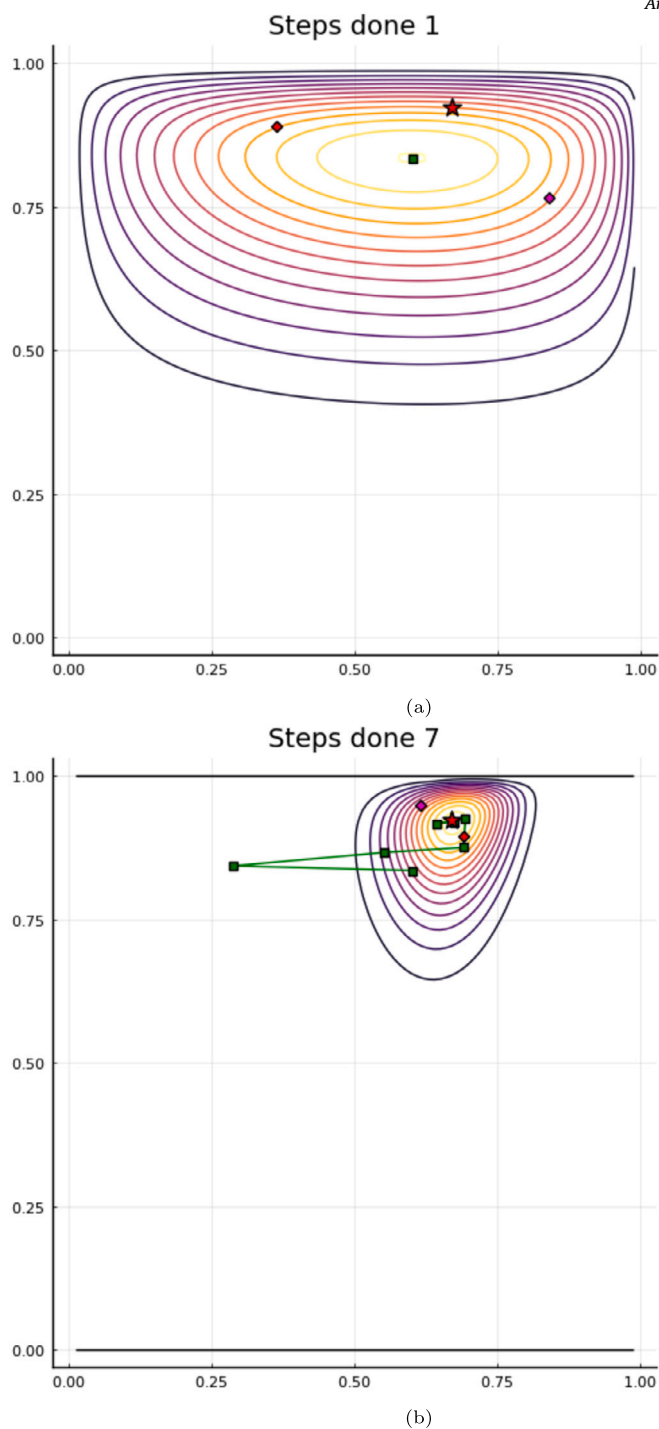


Fig. 2. The inferred synthetic parameter distribution after 1, 7, 20 and 30 interactions in (a)-(d) and the corresponding RMSE, in (e), and RSU, in (f). The red star denotes the true preference, green dots and connecting lines denote the trajectory of the preference estimates before each interaction, the red dot corresponds to the current user preference from the last trial and the magenta dot to the alternative proposal. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

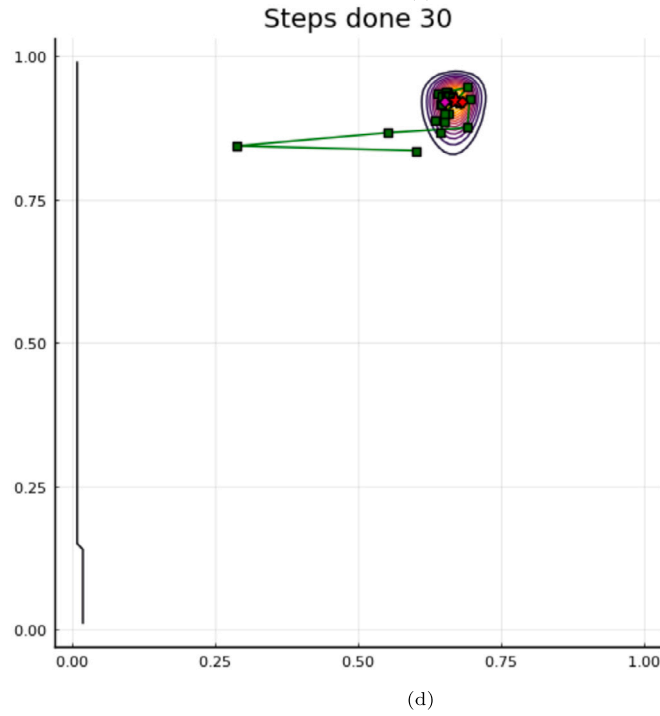
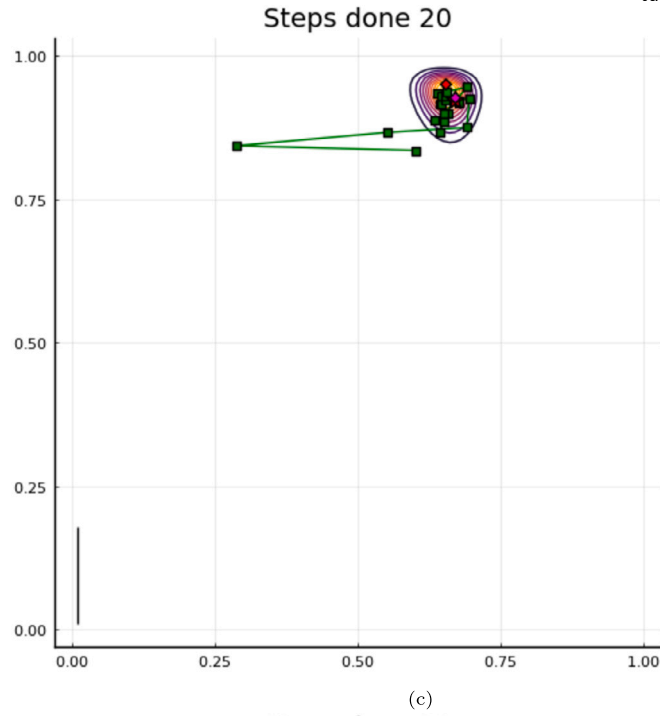
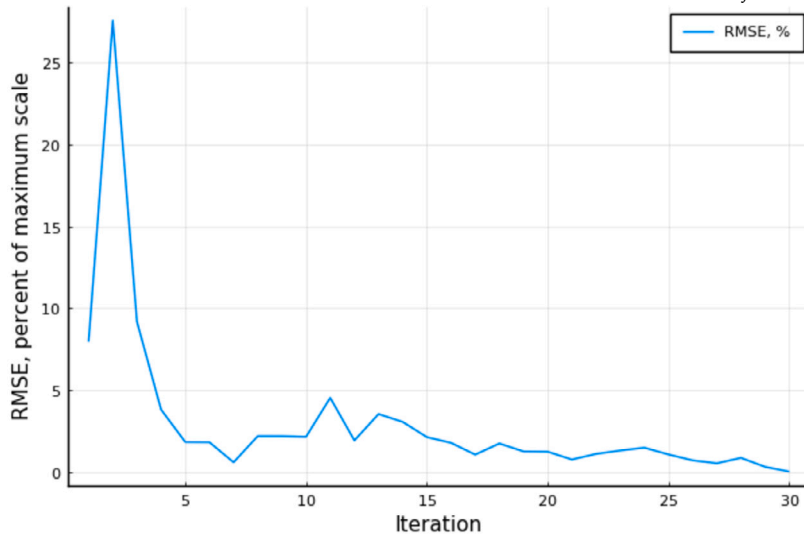


Fig. 2. (continued)

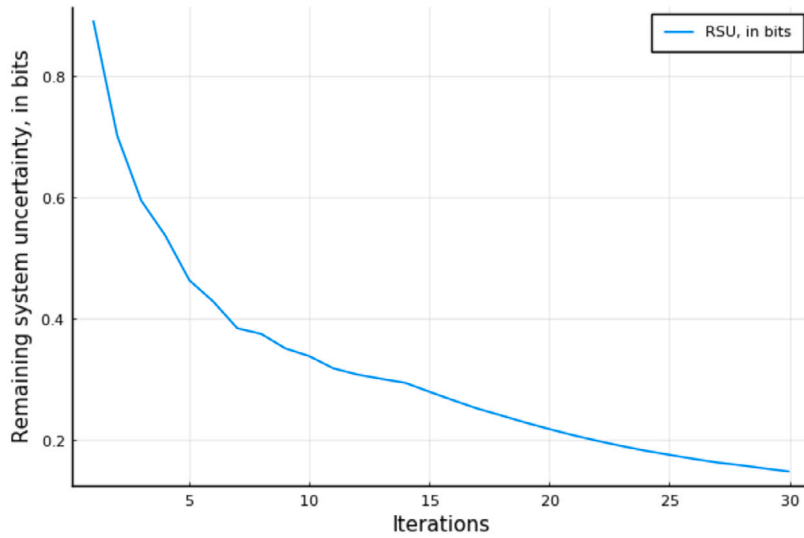
6. Experimental results

6.1. Simulation results

In this section, we present simulation results for a toy example of user preference learning. This example allows us to visualize important aspects of the agent performance. The user model is given by Definition 2.1, where two synthetic parameters have to be learned. We also simulate user responses by sampling them according to the user model, in order to create the whole interaction



(e)



(f)

Fig. 2. (continued)

cycle of the user preference system, see Fig. 1. In our simulations, we know the true values of the user preference function, and our goal is to investigate a number of aspects of the proposed approach. First, we are interested to show feasibility of learning the true preferences by showing that estimated user preferences for synthetic parameters converge to the true user preferences for a simulated user. Moreover, since, in practice, user preferences are unknown, our goal is to validate that RSU provides a good metric for performance monitoring of user preference learning. As a benchmark, we use root-mean-squared error (RMSE), that can be used when the true preference is known.

The numerical experiments were implemented in Python, where probabilistic modeling was realized in probabilistic programming language (PPL), using the Theano-PyMC 1.1.2. In particular, it was used to implement the Hamiltonian Monte-Carlo sampling in Algorithm 2. We used a desktop computer with Intel i7 CPU Core and 32 GB of RAM. Under these conditions, one full interaction step took about 12 sec.

Fig. 2 shows the true optimal parameters, their estimates and the proposals, generated by our agent together with the estimated parameter distribution. Note that as more and more data becomes available, the posterior distribution converges and become very dense around the optimum, and thus the proposals would eventually coincide with or become very close to the true optimum. At the early stage of learning, though, the proposals need not be close to the optimum. At this stage, the exploration process is dominating, where the agent generates proposals that aim at largest reduction of the uncertainty about a user. Indeed, looking at Fig. 2, we see that the alternative proposals at the beginning of the learning process are spread further away from the estimated distribution mean

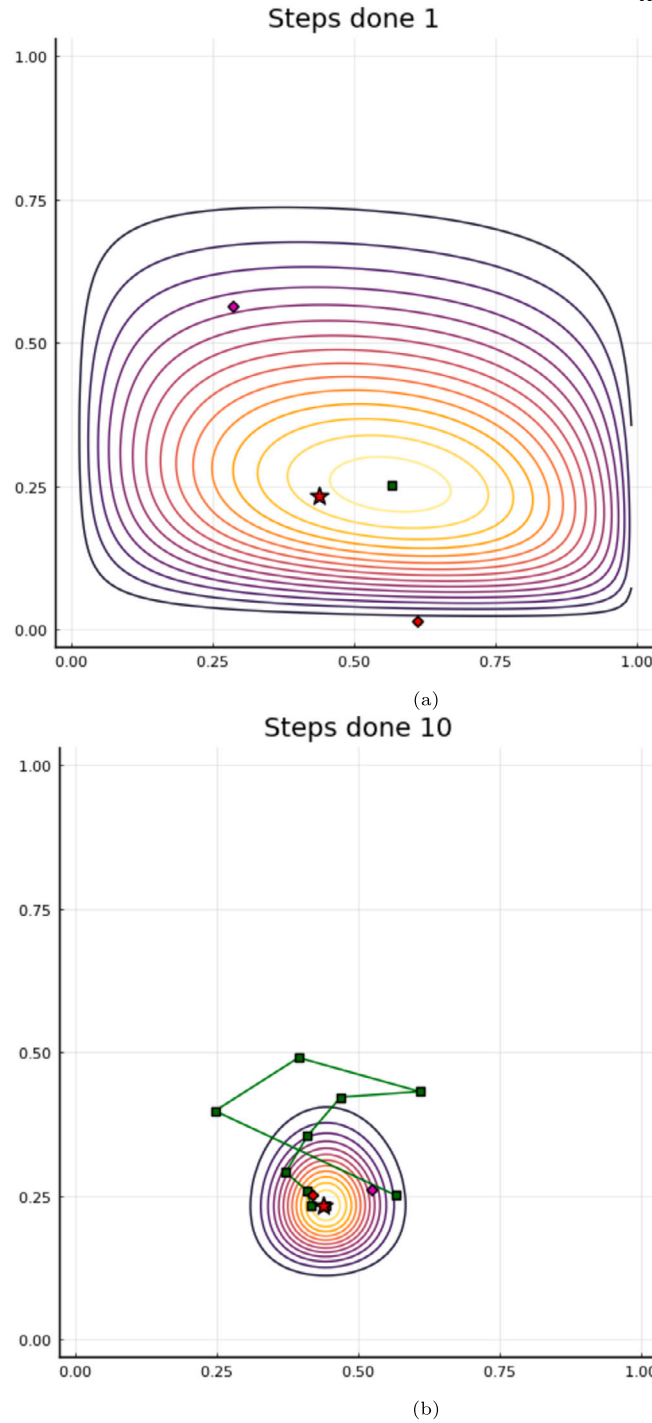


Fig. 3. The inferred synthetic parameter distribution after one, 10, 15 and 30 interactions, in (a)-(d), and the corresponding RMSE, in (e), and RSU, in (f). Notations in (a)-(d) are the same as in Fig. 2.

than at the later stages of learning. In general, we see that the learning procedure of the optimum user preferences converges quickly to the true optimum. Moreover, if we look at RMSE and the agent uncertainty about the user, we see that both RMSE and RSU decrease with time, showcasing quick learning performance of our approach. We run the simulations for 30 interactions, but we see that a good estimate can be achieved with less than 20 interactions. While RSU follows the trend of RMSE, RMSE is more sensitive to inconsistencies in the point estimator. RSU is smoother, since it is a functional of the user response distribution. Moreover, as a functional, it provides more information about the learning behavior than a point estimator. Observe that, in Fig. 3, RSU goes up in

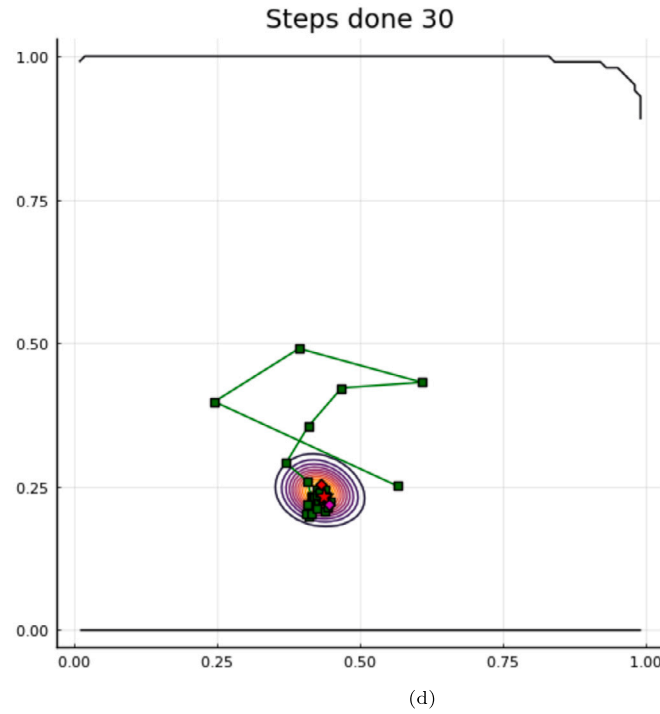
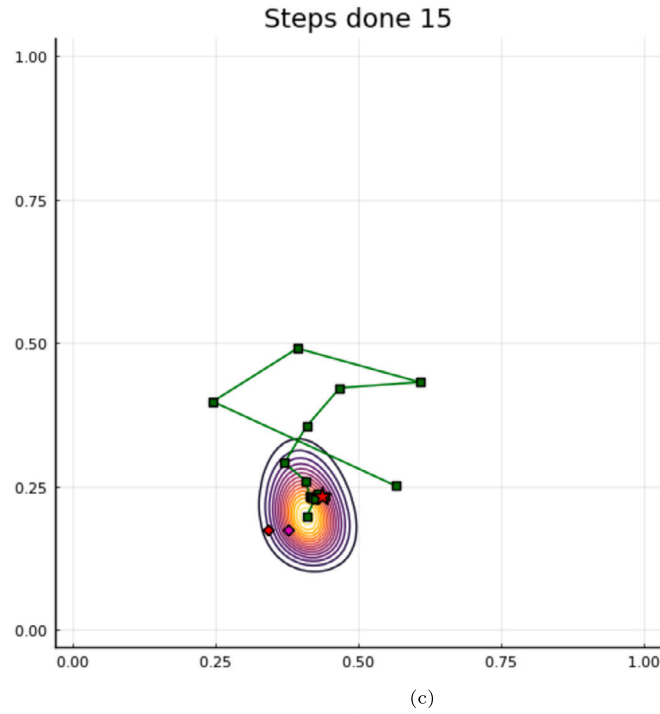
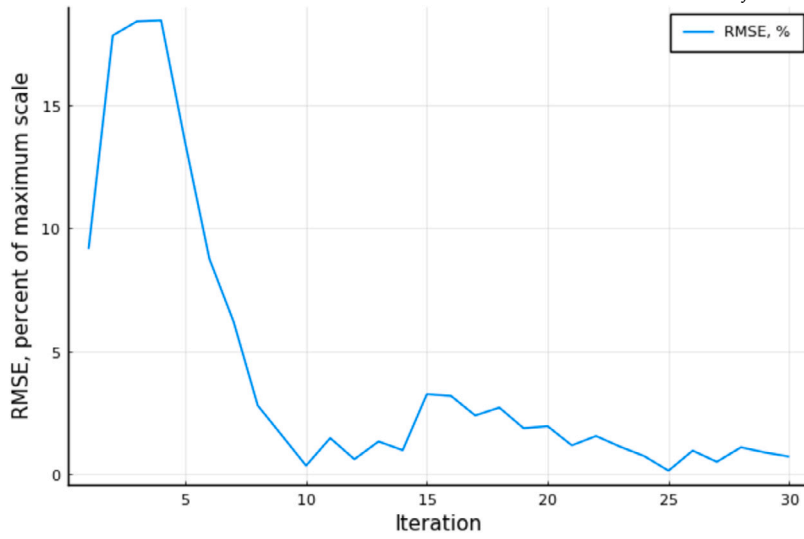


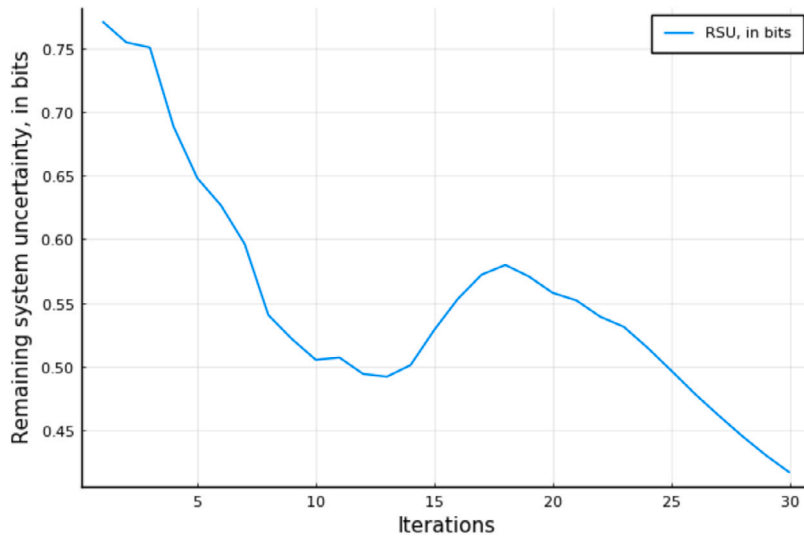
Fig. 3. (continued)

situations when the user becomes less consistent in his responses, but in the end stabilizes as the responses become more reliable and, consequently, the distribution estimate becomes better. This demonstrates that RSU can efficiently be used to monitor the preference learning performance in the field.

Statistical behavior of the performance metrics, RMSE and RSU, is shown in Fig. 4a and Fig. 4b, respectively. These results were obtained by simulating the learning procedure $T = 100$ times, where each time a new true user preference was selected at random, and simulating $n = 30$ interactions in each learning cycle. Note that due to stochastic exploration nature of the agent strategy, RSU



(e)



(f)

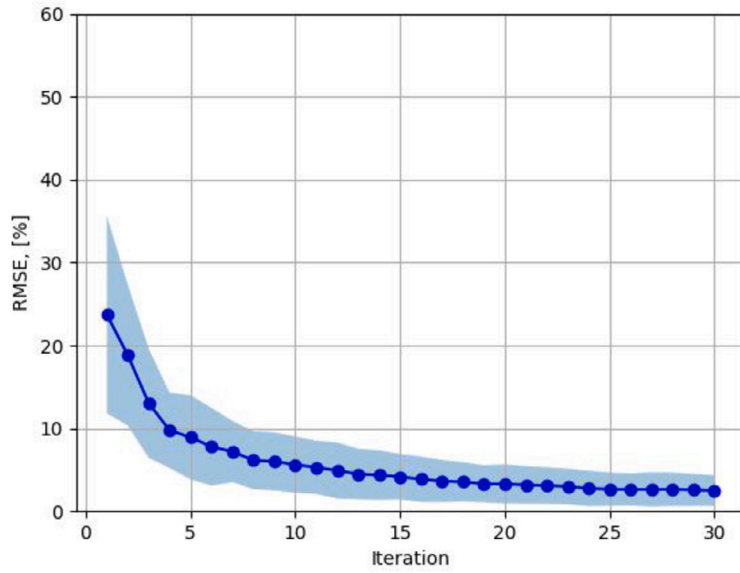
Fig. 3. (continued)

variance increases with the interactions. Indeed, in the beginning of learning reduction of uncertainty is the largest, resulting into small RSU variance. At the later stage, exploration paths can vary with occasional increase in RSU, resulting thus into larger variability.

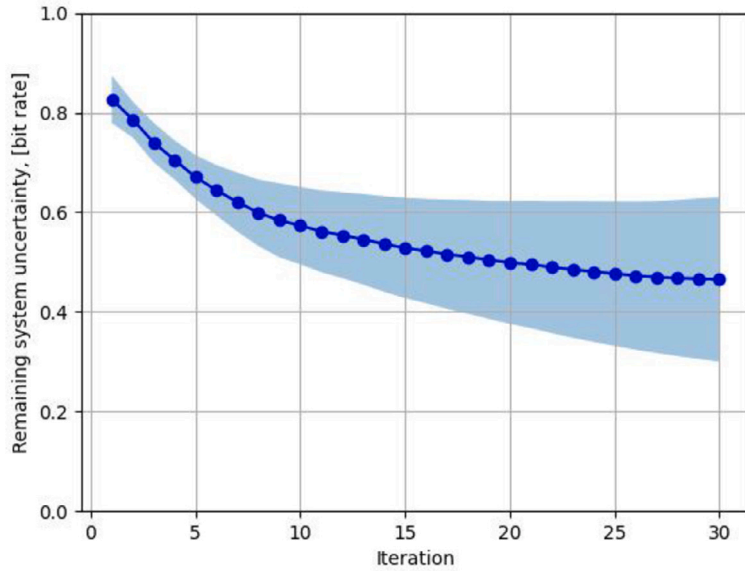
Fig. 5 presents the results of applications of Gaussian processes with different acquisition functions, i.e., expected improvement, upper confidence bound (UCB) and Thomson sampling. Looking at these results, we see that our approach outperforms the resulting methods. After 30 interactions our agent strategy achieves less than 3% RMSE, while the GP-based methods can only achieve from 8 to 11% RMSE on average with rather large variance.

6.2. HA application example

In this section, we discuss the results obtained when we apply the proposed preference learning approach to the problem of HA parameter tuning, discussed in the motivating example. Our goal is to optimize 11 HA parameters. In particular, we focus on the compressor gains for G50 and G80 inputs at five grouped frequency bands, i.e., 250, [500,750], [1000, 1500], [2000, 3000, 4000], 6000 Hz. This grouping is inspired by the sound engineering approach, as these frequency bands are grouped according to their speech-related content, which is relevant for the typical use cases requiring HAs, such as speech, interpersonal communication and music. Fine-tuning specific frequencies according to these principles for individual fricatives is also used by experienced audiologists. The last, 11th, parameter corresponds to the noise suppression level. Given these optimization parameters, we conducted the trial with seven hearing impaired users.



(a) Root-mean-squared error (RMSE)



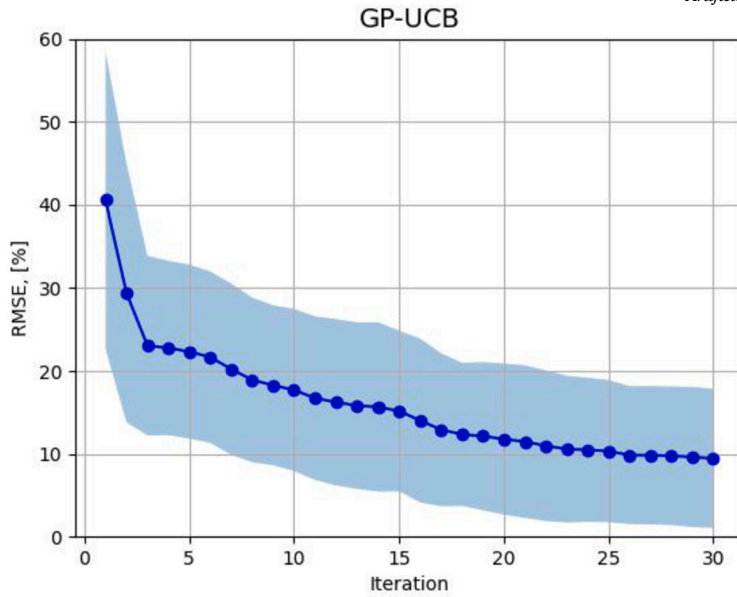
(b) Remaining system uncertainty (RSU)

Fig. 4. RMSE and RSU with 1σ confidence intervals.

In the trial, for the user-agent interactions, the users were presented with two sound samples processed with two sets of parameters. The users have been instructed to think about both their comfort and ability to understand the speech, while evaluating the proposals. The interactions continued until the users were satisfied with the settings and could not perceive the differences in the presented sound samples. We have conducted three study session per user, where three different initial sound settings were used: professionally selected and tuned settings (called “experienced”), settings based on self-measured hearing loss (called “first-fit”), and random settings (called simply “random”).

In order to evaluate the effect of the preference learning, speech recognition task was performed for initial and learned conditions. During this task, the users were presented with 50 non-sense sentences (different for initial and learned conditions) from the BEL sentence corpus [18,19]. The task was intentionally difficult and did not allow to infer possibly missing words from the context.

Fig. 6 shows RSU and the number of steps taken before we conclude that the user has found his/her preferences. Given 11 parameters to optimize, the results look very promising, as it only required the users to take at most 24 steps in order to obtain

Fig. 5. RMSE with 1σ confidence intervals for GP-based agents.**Table 1**
Measured speech improvement, in %.

	Keywords	WER	WIP
Experienced	5.21	5.55	2.18
Self-fit	28.07	19.76	38.33
Random	24.95	13.69	23.48

Table 2
Speech recognition performance.

	Keywords	WER	WIP
Experienced initial	2.1207	0.5040	0.3839
Experienced learned	2.2481	0.4816	0.3934
Self-fit learned	2.5489	0.4085	0.4687
Random learned	2.2713	0.4775	0.3969

satisfying HA settings. Note that the number of steps does not have a dependency on the initial conditions. For Subject 1, e.g., it took 24 steps to fine-tune the parameters, when the subject began with good, professionally selected settings; while Subject 2 only had to perform 11 interactions for the same condition.

Next, we look at the speech recognition performance. While the leaned HA settings are perceived as pleasant and desirable, we are also interested in speech recognition performance delivered by the learned settings. Table 1 presents speech improvement results for different initial conditions. The results are average improvement over 50 sentences, measured as the number of correctly recognized keywords in the sentences; word error rate (WER); and word information preserved (WIP). We observe improved speech recognition for all initial conditions. Clearly, improvement is smallest when initial conditions were already quite good, i.e., selected and fine-tuned by a professional. Nevertheless, even for this condition we achieve both speech recognition improvement and higher sound quality satisfaction by the users.

Finally, using the results presented in Table 2, we analyze the difference in the speech recognition performance, measured as average number of correct keywords (Keywords), and WER and WIP, when different initial conditions were used. We see that the resulting speech recognition performance is comparable for all the conditions and is even better for all but fine-tuned professional settings. Thus, looking at these results, we can conclude that the proposed preference learning approach is a promising technique to be applied in the hearing aid industry.

7. Discussions and conclusions

In this paper, we have studied the problem of active preference learning using pairwise comparison. We have modeled preference learning system as two interacting sub-systems, one representing a user with his preferences for some outcome, characterized by a

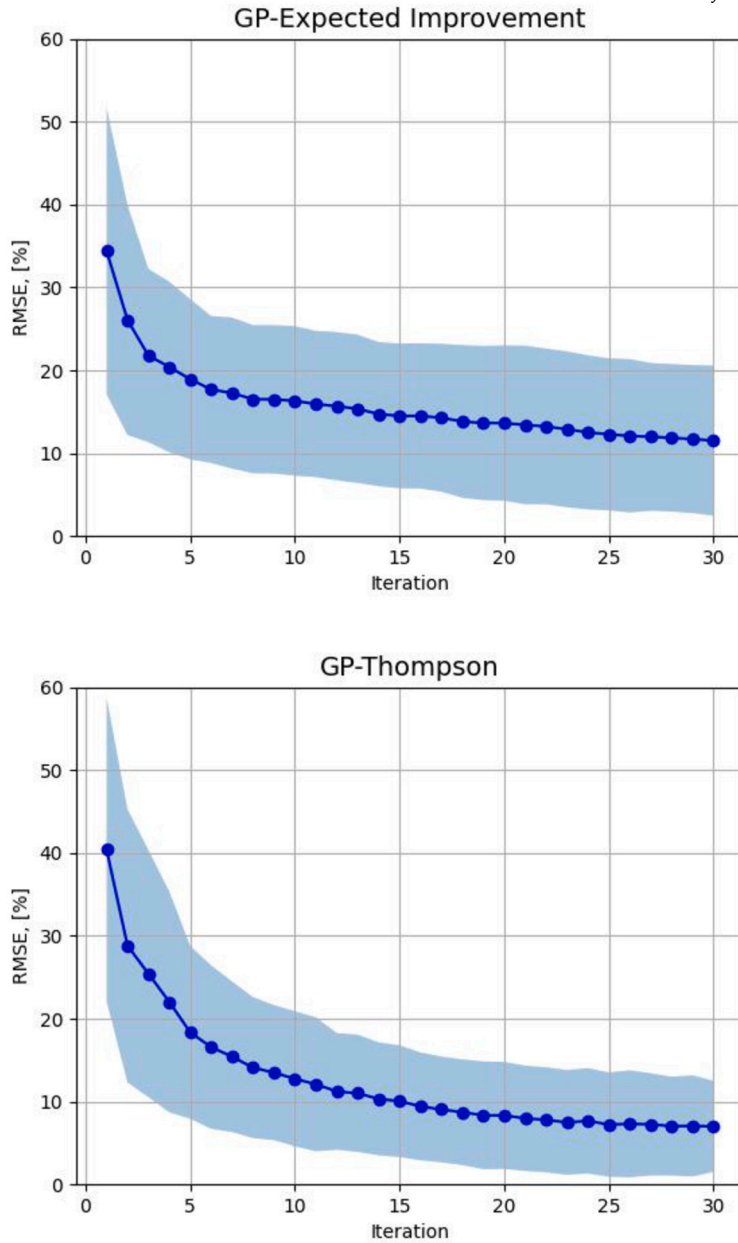


Fig. 5. (continued)

set of parameters, and another one representing the agent that guides the learning process by offering informative trials. In order to monitor the learning behavior and realize the agent strategy for obtaining efficient data from interactions with a user, we have proposed to use the weighted normalized information divergence. This divergence characterizes the agent uncertainty about the user. Optimization of this performance measure results into an optimal strategy for the data collection and guarantees efficient accurate learning of the user preference. Simulation results show that the remaining system uncertainty can be used to monitor the learning behavior of the agent in practice, when the true user preference is unknown. Moreover, we also see that the agent trial generation, based on this metric, results in quick reduction of the search space for the optimal user parameter preferences and their accurate estimation. Also the user study that applied our approach to the practical problem of HA personalization demonstrated quick convergence to good HA settings, which deliver speech improvement and better sound quality.

Our work has a close connection to universal prediction, as well as universal source coding. Here, however, our goal, when assigning a predictive distribution, is user model quality evaluation. Moreover, the problem we study relates to the active inference problem, see e.g. [20]. However, we take somewhat different approach at learning. Instead of putting a focus on designing trials that minimize uncertainty about (future) user responses and thus aiming at supporting the current estimates, we look at the trials that result

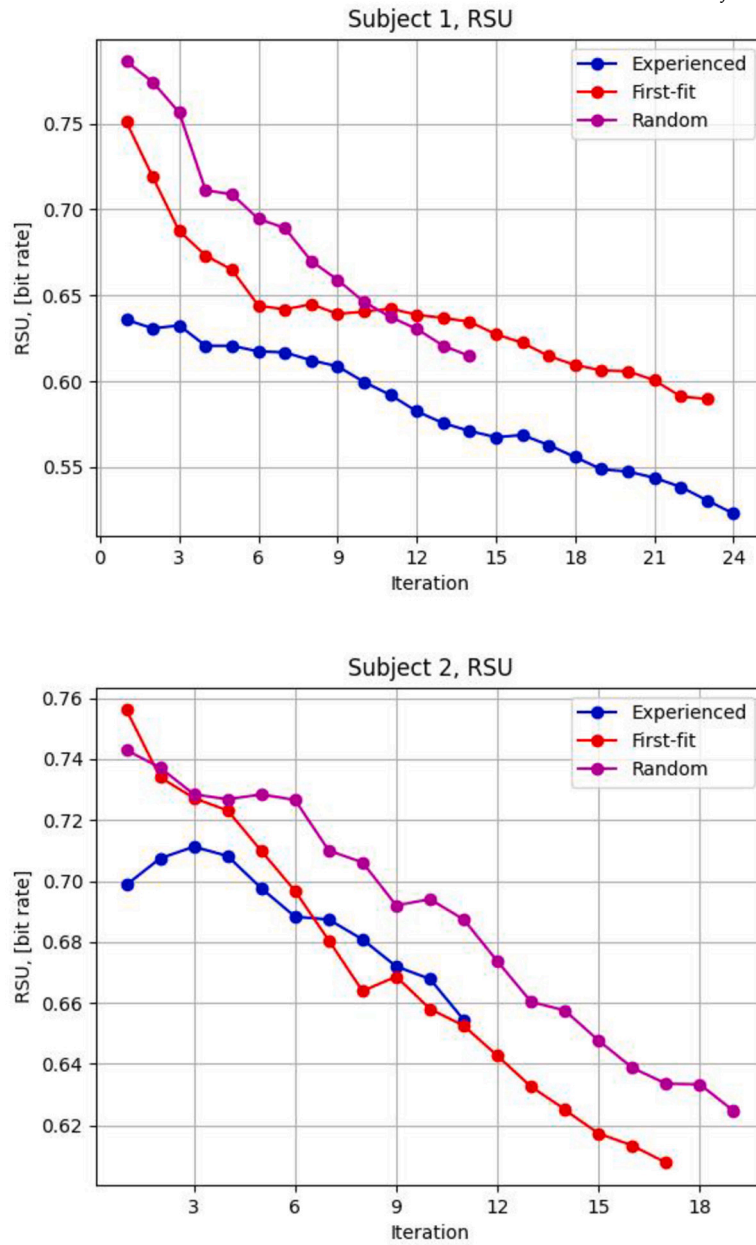


Fig. 6. RSU in HA settings preference learning study.

in the largest uncertainty in the user response and solicit responses to correct the estimate and resolve the uncertainty. Approximate inference, i.e., ADF, plays a role in our work, when we infer the parameters of the user preference function, given actively generated and observed data. Observe that sequential or online nature of the proposed learning approach allows continuous learning of dynamic user preferences in situations when they change over time. Finally, we would like to mention that preference learning has a similar in flavor to reinforcement learning: both are processing the feedback from the environment and have to search for the optimal action strategy. Nevertheless, these two disciplines are conceptually different. First, in reinforcement learning, the goal is known and the reward is formalized using this goal. This is not the case in the preference learning. Next, the state of environment - preference in our case, is unknown and unobservable in preference learning, while reinforcement learning relies on the state observability.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Tanya Ignatenko reports a relationship with GN Advanced Science that includes: employment. Kirill Kondrashov reports a

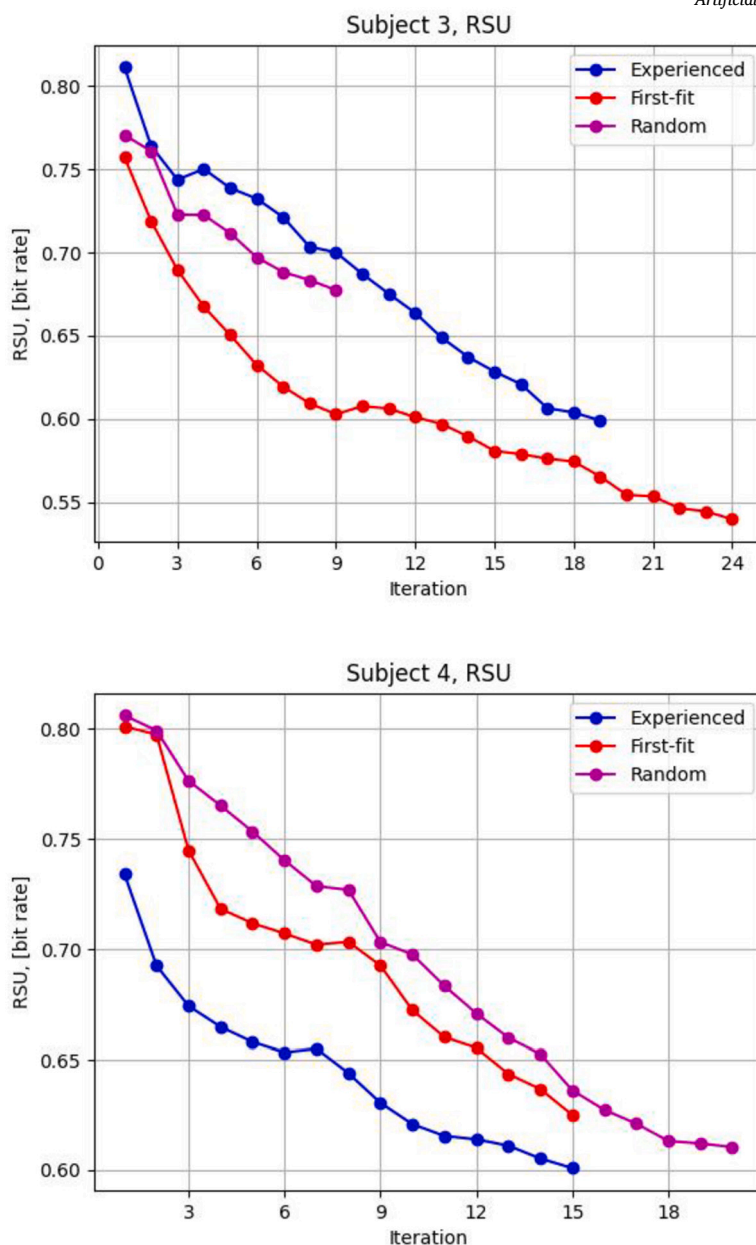


Fig. 6. (continued)

relationship with GN Advanced Science that includes: employment. Marco Cox reports a relationship with GN Advanced Science that includes: employment. Bert de Vries reports a relationship with GN Advanced Science that includes: employment. Tanya Ignatenko has patent pending to GN Hearing A/S. Kirill Kondrashov has patent pending to GN Hearing A/S.

Acknowledgements

We would like to thank Yuri Kirnos for his invaluable help in setting up the user testing system for HA user study; Tao Cui for his help in administrating the study; Greg Olsen and Michalis Papakostas for all their help and support of the user study.

Data availability

The authors do not have permission to share data.

References

- [1] G. Keidser, H. Dillon, M. Flax, T. Ching, S. Brewer, The NAL-NL2 prescription procedure, *Audiol. Res.* 1 (2011).
- [2] M. Kendall, J.D. Gibbons, Rank Correlation Methods, 5th edition, A Charles Griffin Title, 1990.
- [3] E. Brochu, N.d. Freitas, A. Ghosh, Active preference learning with discrete choice data, in: Proceedings of the 20th International Conference on Neural Information Processing Systems, NIPS'07, Curran Associates Inc., Red Hook, NY, USA, 2007, pp. 409–416.
- [4] E. Brochu, V.M. Cora, N. de Freitas, A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning, arXiv:1012.2599, 2010, Preprint.
- [5] J. González, Z. Dai, A. Damianou, N.D. Lawrence, Preferential Bayesian optimization, in: D. Precup, Y.W. Teh (Eds.), Proceedings of the 34th International Conference on Machine Learning, in: Proceedings of Machine Learning Research, vol. 70, PMLR, 2017, pp. 1282–1291, <https://proceedings.mlr.press/v70/gonzalez17a.html>.
- [6] P. Mikkola, M. Todorovic, J. Järvi, P. Rinke, S. Kaski, Projective preferential Bayesian optimization, in: Proc. ICML 2020, the 37th International Conference on Machine Learning, vol. 119, PMLR, 2019, pp. 6884–6892.
- [7] M. Cox, B. de Vries, A parametric approach to Bayesian optimization with pairwise comparisons, in: NIPS Workshop on Bayesian Optimization (BayesOpt 2017), Long Beach, USA, December 9, 2017, 2017.
- [8] P. Hennig, C.J. Schuler, Entropy search for information-efficient global optimization, arXiv, arXiv:1112.1217 [abs], 2012.
- [9] J.M. Hernández-Lobato, M.W. Hoffman, Z. Ghahramani, Predictive entropy search for efficient global optimization of black-box functions, in: Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 1, NIPS '14, MIT Press, Cambridge, MA, USA, 2014, pp. 918–926.
- [10] N. Merhav, M. Feder, Universal prediction, *IEEE Trans. Inf. Theory* 44 (6) (1998).
- [11] W.R. Ashby, The Measurement of Values. By L. L. Thurstone, University of Chicago Press, Chicago, 1959, p. 322; *J. Ment. Sci.* 105 (441) (1959) 1127, <https://doi.org/10.1192/bjp.105.441.1127-a>, publisher: Cambridge University Press.
- [12] T.M. Cover, E. Ordentlich, Universal portfolios with side information, *IEEE Trans. Inf. Theory* 42 (2) (1996) 348–363.
- [13] T. Matsushima, H. Inazumi, S. Hirasawa, A class of distortionless codes designed by Bayes decision theory, *IEEE Trans. Inf. Theory* 37 (5) (1991) 1288–1293.
- [14] N. Merhav, M. Feder, A strong version of the redundancy-capacity theorem of universal coding, *IEEE Trans. Inf. Theory* 41 (3) (1995) 714–722.
- [15] M. Feder, N. Merhav, Hierarchical universal coding, *IEEE Trans. Inf. Theory* 42 (5) (1996) 1354–1364, <https://doi.org/10.1109/18.532877>.
- [16] M. Opper, A Bayesian Approach to on-Line Learning, Cambridge University Press, USA, 1999, pp. 363–378.
- [17] S. Solla, O. Winther, Optimal Erceptron Learning: as Online Bayesian Approach, Cambridge University Press, USA, 1999, pp. 379–398.
- [18] E.R. O'Neill, M.N. Parke, H.A. Kreft, A.J. Oxenham, Recordings and text for new nonsense bel sentence corpus, Data Repository for the University of Minnesota, <https://doi.org/10.13020/tdf8-6p56>.
- [19] E.R. O'Neill, M.N. Parke, H.A. Kreft, A.J. Oxenham, Development and validation of sentences without semantic context to complement the basic English lexicon sentences, *J. Speech Lang. Hear. Res.* 63 (11) (2020) 3847–3854, https://doi.org/10.1044/2020_JSLHR-20-00174, https://pubs.asha.org/doi/pdf/10.1044/2020_JSLHR-20-00174.
- [20] K. Friston, The free-energy principle: a rough guide to the brain?, *Trends Cogn. Sci.* 13 (7) (2009) 293–301, <https://doi.org/10.1016/j.tics.2009.04.005>.