Proceedings of the Eleventh AAAI Conference on Human Computation and Crowdsourcing (HCOMP 2023)

A Cluster-Aware Transfer Learning for Bayesian Optimization of Personalized Preference Models

Haruto Yamasaki[1], Masaki Matsubara[1], Hiroyoshi Ito[1], Yuta Nambu[2], Masahiro Kohjima[2], Yuki Kurauchi[2], Ryuji Yamamoto[2], Atsuyuki Morishima[1]

[1] University of Tsukuba, Japan
[2] NTT Human Informatics Laboratories
s2321716@u.tsukuba.ac.jp, masaki@slis.tsukuba.ac.jp, ito@slis.tsukuba.ac.jp, yuta.nambu@ntt.com, masahiro.kohjima@ntt.com, yuki.kurauchi@ntt.com, ryuji.yamamoto@ntt.com, mori@slis.tsukuba.ac.jp

Abstract

Obtaining personalized models of the crowd is an important issue in various applications, such as preference acquisition and user interaction customization. However, the crowd setting, in which we assume we have little knowledge about the person, brings the cold start problem, which may cause avoidable unpreferable interactions with the people. This paper proposes a cluster-aware transfer learning method for the Bayesian optimization of personalized models. The proposed method, called Cluster-aware Bayesian Optimization, is designed based on a known feature: user preferences are not completely independent but can be divided into clusters. It exploits the clustering information to efficiently find the preference of the crowds while avoiding unpreferable interactions. The results of our extensive experiments with different data sets show that the method is efficient for finding the most preferable items and effective in reducing the number of unpreferable interactions.

Introduction

Obtaining personalized models of people is indispensable in today's society to increase company sales and customer satisfaction and yield better user experience (Anshari et al. 2019; Tyrväinen, Karjaluoto, and Saarijärvi 2020; Adaji 2017). Today, we see more and more applications that require personalized preference models of people in online crowd settings; the obtained preference models are used for the creation of avatars for online games (preferred attributes of components) and individual optimization of UIs (preferred shapes and colors, etc.), and better task assignment (preferred condition of the tasks) (Yao, Yang, and Xu 2022; Gajos, Weld, and Wobbrock 2010).

However, in the crowd setting, we often have little knowledge about the person in the beginning. Bayesian optimization using Gaussian process regression, a technique for black-box function optimization, can be used to obtain personalized models (Brochu 2010; Korzepa et al. 2020) to deal with the situation, by regarding an individual's preference as a black-box function. Bayesian optimization is an efficient algorithm compared to conventional optimization methods such as grid search and random search (Bergstra and Bengio 2012; Liashchynskyi and Liashchynskyi 2019), as it continuously searches for the next point to be evaluated while considering both exploration and exploitation.

A general problem of Bayesian optimization is the cold start (Wang et al. 2022), i.e., the inference does not work well until we gather sufficient information. In addition, the cold start can be harmful, particularly for human preference models, because it can show unpreferable items to people during the optimization process, which may cause an unpleasant experience. For example, when estimating preferred images and sounds, the respondent may be presented with images and sounds that he or she does not like, which may cause discomfort to the respondent.

To deal with the cold start, several methods based on the transfer learning approach have been proposed (Bardenet et al. 2013; Marco et al. 2017). In our context, transfer learning addresses the cold start for a personalized preference model by referencing others' models. Existing transfer learning methods, however, cannot apply to the problem of obtaining personalized preference models in the crowd setting due to their assumptions that are not compatible with the problem. For example, they are too computationally expensive (Bardenet et al. 2013; Swersky, Snoek, and Adams 2013; Schilling, Wistuba, and Schmidt-Thieme 2016; Papež and Quinn 2022) for real-time interactions with people or require free anytime access to any of source tasks (other people in our setting) during the optimization process (Marco et al. 2017; Wistuba, Schilling, and Schmidt-Thieme 2018), and they do not consider exploiting the transfer learning to avoid showing unpreferred items to people, which is an important issue for interactions to obtain personalized preference models.

This paper proposes a cluster-aware transfer learning method for the Bayesian optimization of personalized models. The proposed method, called Cluster-aware Bayesian Optimization (CBO), is designed based on a known feature that is specific to humans' preference models; user preferences are not nesessary independent but can be divided into clusters (Rentfrow and Gosling 2003; Silva et al. 2019; Ohn-Bar et al. 2018). CBO exploits the clustering information to efficiently find the preference of the crowds while avoiding

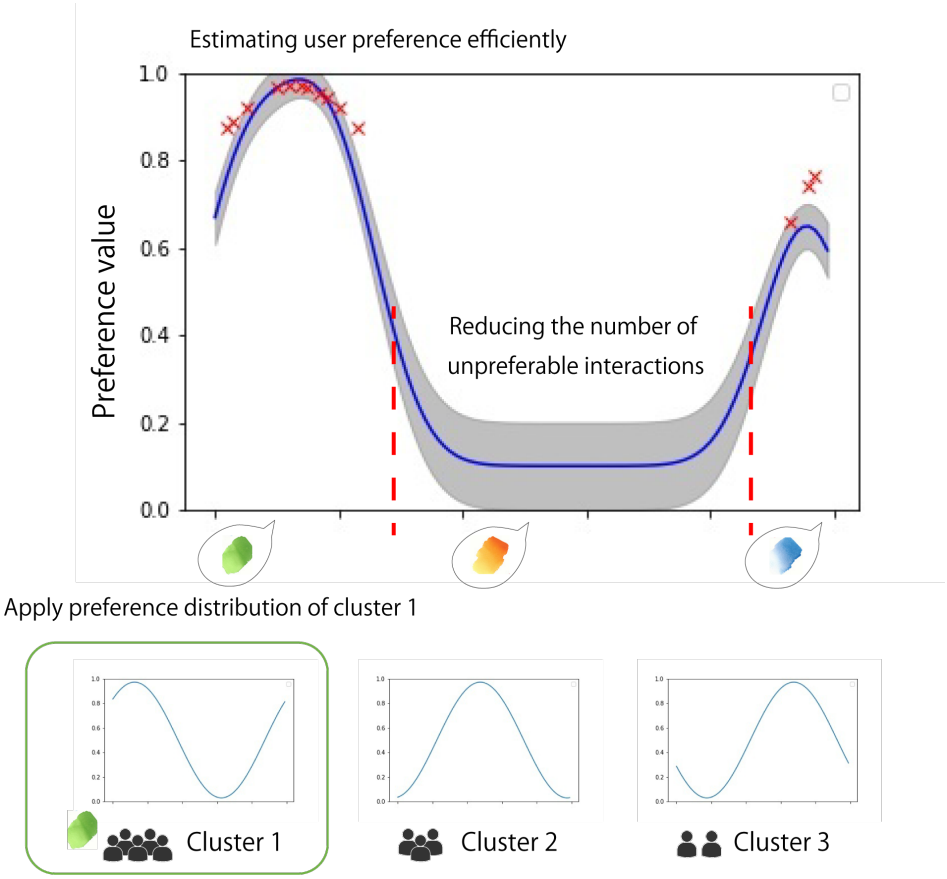


Figure 1: Proposed method: Cluster-aware Bayesian optimization (CBO). CBO estimates the user's preference distribution with reference to the distributions of clusters. CBO approach can efficiently find the most preferable items while avoiding the less-preferred items by utilizing the information of what cluster the human belongs to.

unpreferable interactions.

Contributions and Findings. To summarize, our research question is whether cluster-aware transfer learning is effective for the cold start problem of Bayesian optimization of personalized preference models in the crowd setting. Our contributions are as follows: (1) We formally define cluster-based transfer learning for Bayesian optimization for quickly finding the most preferable items while reducing the number of unpreferable interactions. (2) We conducted extensive experiments with synthesized and three real-world data sets we collected through crowdsourcing and revealed the detailed behavior of the cluster-aware Bayesian optimization.

We found the followings: (1) Transfer learning in CBO works well in that it efficiently identifies the preference models while avoiding presenting solutions with low preference, especially at the beginning of the process (2) There is a good hyper-parameter setting strategy for CBO that works with a variety of data sets. (3) Since our transfer learning technique changes the optimization problem into the problem of identifying the appropriate cluster, it becomes superior to the optimization without transfer learning, as the number of dimensions of the search space increases.

Related Work

Bayesian Optimization Using Transfer Learning

Various transfer learning methods have been developed within the framework of Bayesian optimization to further improve the data efficiency of optimization by utilizing already optimized data (source task) when optimizing some new data (target task). For example, initializing the exploration, transferring information to a Gaussian process, transferring information to an acquisition function, and so on (Feurer, Springenberg, and Hutter 2015; Bardenet et al. 2013; Wistuba, Schilling, and Schmidt-Thieme 2018).

Feurer et al. proposed a technique that queries the points with good results in the source task in the early stages of optimization of the target task (Feurer, Springenberg, and Hutter 2015). However, their method is not applicable to the cold start problem considered in this study because they need a known set of tasks (clusters in our context). Even if they knew it, they would not avoid low-preference items.

Some transfer learning methods have been proposed to transfer information to Gaussian processes, but they generally require $\mathcal{O}(n^3)$ as the computational complexity, where n is the number of all data in the source and target tasks (Bar-

denet et al. 2013; Swersky, Snoek, and Adams 2013; Schilling, Wistuba, and Schmidt-Thieme 2016; Papež and Quinn 2022), or require free access to the source task (Marco et al. 2017). Therefore, these methods are not applicable in our setting, which aims to optimize the preferences of individual respondents through interactions with them alone while utilizing a large amount of source data.

Wistuba et al. proposed a method that transfers information to the acquisition function to freely control the importance of the transferred information (Wistuba, Schilling, and Schmidt-Thieme 2018). However, the method is unsuitable for this study because they require free access to the source task.

Bayesian Optimization for Safe Exploration

Sui et al. proposed a method called SafeOpt that performs optimization while avoiding exploration that returns values below a certain threshold with high probability by utilizing the Lipschitz continuity of the Gaussian process (Sui et al. 2015). The assumption of SafeOpt is that there is at least one point above the threshold in everyone's preference model. In our setting, not only do we not know where such a point is, it generally does not exist at all. In addition, this method requires a lot of interaction with the target user and may need repeated evaluations of similar points since it carefully moves through the exploration space so as not to explore points with low ratings. SafeOpt intends to be safe while sacrificing the efficiency, which does not fit our goal of finding the preferences of humans efficiently.

Prerequisite Knowledge

Gaussian Process Regression

In this section, we describe the Gaussian process regression used within CBO. It is a model that estimates the function $y = f(\mathbf{x})$ from the input variable $\mathbf{x}$ to the output variable y (Rasmussen and Williams 2004). Unlike ordinary regression, the variance at each point $\mathbf{x}$ is also obtained so that the uncertainty at every point of the estimated function can also be estimated simultaneously.

The definition of a Gaussian process is as follows. For any natural number N, f follows a Gaussian process when the output vector $\mathbf{f} = (f(\mathbf{x}_1), f(\mathbf{x}_2), ..., f(\mathbf{x}_N))$ corresponding to the input $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N \in \mathcal{X}$ follows a Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \mathbf{K})$ with $\boldsymbol{\mu} = (\mu(\mathbf{x}_1), \mu(\mathbf{x}_2), ..., \mu(\mathbf{x}_N))$ as the mean and $\mathbf{K}$ with $K_{nn'} = k(\mathbf{x}_n, \mathbf{x}_{n'})$ as elements as the covariance matrix. This is expressed by the following equation.

$$f \sim GP(\boldsymbol{\mu}(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')) \tag{1}$$

Then, given N observations, that is, N pairs $\mathcal{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_N, y_N)\}$ of inputs $\mathbf{x} \in \mathcal{X}$ and outputs $y \in \mathbb{R}$ (the mean of y is μ). When the relationship $y = f(\mathbf{x})$ exists between $\mathbf{x}$ and y, and this function f is regarded as being generated from a Gaussian process

$$f \sim GP(\mu(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$$

the relationship between $\mathbf{x}$ and y can be treated by a Gaussian process regression model.

Given an observation $\mathcal{D}$, the predictive distribution $p(y^*|\mathbf{x}^*, \mathcal{D})$ of y^* corresponding to the unknown data $\mathbf{x}^*$ is expressed by the following equation.

$$p(y^*|\mathbf{x}^*, \mathcal{D}) = \mathcal{N}(\mathbf{k}_*^T \mathbf{K^{-1}} \mathbf{y}, k_{**} - \mathbf{k}_*^\mathbf{T} \mathbf{K^{-1}} \mathbf{k}_*) \tag{2}$$

$\mathbf{k}_*$ and k_{**} are as follows.

$$\mathbf{k}_* = (k(\mathbf{x}_*, \mathbf{x}_1), k(\mathbf{x}_*, \mathbf{x}_2), \cdots, k(\mathbf{x}_*, \mathbf{x}_N))^T \tag{3}$$

$$k_{**} = k(\mathbf{x}^*, \mathbf{x}^*) \tag{4}$$

In this case, the expected value and variance of the predictive distribution are expressed by the following equations, respectively.

$$\mathbb{E}[y^*|\mathbf{x}^*, \mathcal{D}] = \mathbf{k}_*^T \mathbf{K^{-1}} \mathbf{y} \tag{5}$$

$$\mathbb{V}[y^*|\mathbf{x}^*, \mathcal{D}] = k_{**} - \mathbf{k}_*^\mathbf{T} \mathbf{K^{-1}} \mathbf{k}_* \tag{6}$$

Bayesian Optimization

Bayesian optimization (Frazier 2018; Brochu, Cora, and De Freitas 2010) is the problem of finding $\mathbf{x}_{opt} = \arg\max_{x \in \mathcal{X}} f(\mathbf{x})$ while minimizing the number of evaluations of the black-box function $f(\mathbf{x})$. Since the uncertainty of the function must also be taken into account to minimize the number of evaluations, Bayesian optimization often assumes that the function is sampled from a Gaussian process. Bayesian optimization using Gaussian process regression is performed by repeating the process of sampling a new point that considers both exploration and exploitation using an acquisition function and obtaining the function value by evaluating it. There are several acquisition functions, each of which selects an item in various criteria. For example, probability of improvement (PI), expected improvement (EI), and Gaussian process upper-confidence bound (GP-UCB) are commonly used as the acquisition function (Snoek, Larochelle, and Adams 2012). The value of the acquisition function is calculated for all points, and the point with the largest value of it is selected in each iteration.

In the proposed cluster-aware Bayesian optimization, we extend a generic acquisition function GP-UCB. Let $\mathbb{E}[y^*|\mathbf{x}^*, \mathcal{D}^{(t)}]$ and $\mathbb{V}[y^*|\mathbf{x}^*, \mathcal{D}^{(t)}]$ be the expected value and variance of the distribution in point $\mathbf{x}^*$ created by Gaussian process regression for the points $\mathcal{D}^{(t)}$ selected up to the first t iterations and their function values, respectively. In this case, the acquisition function GP-UCB that determines the point to be evaluated at the $t+1$ iteration is expressed by the following equation.

$$acquisition(\mathbf{x}^*) = \mathbb{E}[y^*|\mathbf{x}^*, \mathcal{D}^{(t)}] \ + \ \alpha_1 \sqrt{\mathbb{V}[y^*|\mathbf{x}^*, \mathcal{D}^{(t)}]} \tag{7}$$

The α_1 is a weight for the standard deviation term and which is a hyper-parameter. When α_1 is large, the value of the standard deviation term, which represents the uncertainty of the function, is reflected in the acquisition function to a greater extent, thus making it easier to sample a point for the exploration. When α_1 is small, the value of the expected value term, which represents the function's evaluation value, is reflected in the acquisition function to a greater extent, thereby making it easier to sample a point for exploitation.

Proposed Method

Problem Setting

In this study, we assume a set of the respondents $\mathcal{W} = \{w_1, ..., w_{|\mathcal{W}|}\}$ each of which returns the function value y for the given point $\mathbf{x}$. We define the obtained data from a respondent as $\mathcal{D}_w = \{(\mathbf{x}_i, y_i)\}$, and the all obtained data as $\mathcal{D} = \{\mathcal{D}_w\}_{w \in \mathcal{W}}$. Especially, we note $\mathcal{D}_w^{(t)} = \{(\mathbf{x}_i, y_i)\}$ as a set of obtained data from a respondent w from 1st to t-th iterations. We define the predictive distribution of a respondent w as $p(y^* \mid \mathbf{x}^*, \mathcal{D}_w)$ of which the mean and the covariance matrix is calculated by $\mathcal{D}_w$, and the expected value and variance are $\mathbb{E}[y^* \mid \mathbf{x}^*, \mathcal{D}_w]$ and $\mathbb{V}[y^* \mid \mathbf{x}^*, \mathcal{D}_w]$, respectively.

Proposed Method: Cluster-Aware Bayesian Optimization (CBO)

This section expresses our proposed method CBO, which exploits the preference of clusters of humans to efficiently find their preference while avoiding unpleasant items.

CBO firstly divides people whose preferences have already been estimated into several clusters based on the similarity of their preferences. After estimating their clusters, CBO applies the average of the expected values of the distribution of people in the appropriate cluster (call it *c-expected value*) to the generic acquisition function GP-UCB. Using the information from the clusters allows exploration according to the respondent's preferences, which is thought to allow for efficient optimization while avoiding interactions that are less preferred by the respondent.

Since the importance of the information from the clusters should be changed during the optimization process, our proposed method transfers the information from the clusters to the acquisition function.

Clustering People CBO uses the k-means method to divide people into clusters. We note n_c as a number of clusters. The function $\mu_k(\mathbf{x})$ is the centroid of each cluster, and $\mathbf{r}_w = (r_{w1}, \cdots, r_{wn_c})^\top$ is a vector that expresses to which cluster an respondent w belongs after t iterations. The k-means clustering is a minimization problem for the following loss function $\mathcal{L}$.

$$\mathcal{L} = \sum_{w \in \mathcal{W}} \sum_{k \in n_c} r_{wk} \int \| \mathbb{E}[y^* \mid \mathbf{x}^*, \mathcal{D}_w] - \mu_k(\mathbf{x}) \|_2^2 \, d\mathbf{x} \tag{8}$$

$\mathbf{r}_w$ is determined by the following equation.

$$r_{wj} = \begin{cases} 1 & j = \arg\min_k \int \| \mathbb{E}[y^* \mid \mathbf{x}^*, \mathcal{D}_w] - \mu_k(\mathbf{x}) \|_2^2 \, d\mathbf{x} \\ 0 & \text{Otherwise.} \end{cases} \tag{9}$$

That is, for each respondent, the cluster $\mathcal{C}_k$ to which they belong is determined by the following two equations.

$$\{\mathbf{r}_w\}_{w \in \mathcal{W}}, \{\mu_k\}_{k \in [n_c]} = \arg\min_{\mathbf{r}_w, \mu_k} \mathcal{L} \tag{10}$$

$$\mathcal{C}_k = \{w \in \mathcal{W} \mid r_{wk} = 1\} \tag{11}$$

A centroid $\mu_k(\mathbf{x})$ is calculated as an expected value of a k-th cluster which is defined by the following equation:

$$\mu_k(\mathbf{x}^*) = \frac{1}{|\mathcal{C}_k|} \sum_{w \in \mathcal{C}_k} \mathbb{E}[y^* \mid \mathbf{x}^*, \mathcal{D}_w]. \tag{12}$$

We call $\mu_k(\mathbf{x}^*)$ as *c-expected value*. The number of clusters n_c is a hyper-parameter in the k-means. Note that, the number of clusters can be automatically determined by using x-means (Pelleg, Moore et al. 2000).

How to Apply the C-Expected Value When estimating a respondent's preference, the c-expected value of the cluster to which a respondent belongs is added to an acquisition function in CBO. Acquisition functions that take into account cluster information are considered to be more likely to fall into locally optimal solutions than acquisition functions not considering it. Therefore, GP-UCB was selected as the generic acquisition function for our method, as it provides the most flexible balance between search and utilization.

We note $\mathbf{r}_w^{(t)}$ as the cluster assignment at t iterations which is estimated by using the equation 9 with recalculated $\mathbb{E}[y^* \mid \mathbf{x}^*, \mathcal{D}_w^{(t)}]$.

The acquisition function of CBO is expressed by the following equation using the expected value $\mathbb{E}[y^* | \mathbf{x}^*, \mathcal{D}_w^{(t)}]$ and variance $\mathbb{V}[y^* | \mathbf{x}^*, \mathcal{D}_w^{(t)}]$ estimated at time t and the vector $\mathbf{r}_w^{(t)}$.

$$\begin{aligned} acquisition(\mathbf{x}^*, w, t) = \; & \mathbb{E}[y^* \mid \mathbf{x}^*, \mathcal{D}_w^{(t)}] \\ & + \; \alpha_1^{(t)} \sqrt{\mathbb{V}[y^* \mid \mathbf{x}^*, \mathcal{D}_w^{(t)}]} \\ & + \; \alpha_2^{(t)} \sum_{k \in [n_c]} r_{wk}^{(t)} \, \mu_k(\mathbf{x}^*) \end{aligned} \tag{13}$$

where $\alpha_1^{(t)}$, and $\alpha_2^{(t)}$ are the hyper-parameters. To control the effect of the c-expected values and the variance, we dynamically change the parameters according to the number of iterations. We discuss a good strategy for changing the parameters in the experiment.

The point that maximizes the acquisition function 13 is then presented to w at t-th iteration. In other words, the following point $\mathbf{x}_t$ is presented.

$$\mathbf{x}_t = \arg\max_{\mathbf{x}^* \in \mathcal{X}} \; acquisition(\mathbf{x}^*, w, t) \tag{14}$$

Compared to the GP-UCB (equation 7), we can see that equation 13 has an additional term for the c-expected value. There are two hyper-parameters, α_1 and α_2. By tuning these values, we can adjust what the points to be acquired. It is no different from GP-UCB in that when α_1 is set to a large value, points for exploration are more likely to be chosen. When α_2 is set to a large value, points that emphasize the c-expected value, that is, points for exploitation are more likely to be selected. When the cluster to which a respondent belongs is accurately determined, it is expected that the c-expected value term can be used to sample a point with a high rating even when the number of iterations is small.

The entire flow of the method is shown in Algorithm 1. This Algorithm 1 is not for a single respondent, but for a set of respondents. There is a step of repeated recalculation of the expected value, variance, and cluster to which a respondent belongs for each iteration and a step of repeated recalculation of the clustering for each respondent.

Notation	Description
$\mathcal{W}$	respondent set. A respondent is $w \in \mathcal{W}$.
$\mathcal{D}$	Data set. A respondent's data is $\mathcal{D}_w = \{(\mathbf{x}_i, y_i)\}$
$\mathcal{C}$	Set of clusters. $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_{n_c}\}$
$\mu_k(\mathbf{x})$	c-expected value (Centroid function) of each cluster.
n_c	Number of clusters.
$\mathbf{r}_w$	A vector expresses to which cluster a respondent w belongs.
$\mathbf{x}$	Solution's feature.
y	Function value of each solution.
T	Number of queries per respondent.

Table 1: Notations used in the problem setting

Algorithm 1: Entire flow of CBO

Input: A set $\mathcal{W}$ of respondents, T
Output: $\mathcal{D}$

$\mathcal{D} \leftarrow \emptyset$
for $w \in \mathcal{W}$ **do**
  for $t \in [T]$ **do**
    $\mathbf{x}_t = \underset{\mathbf{x}^* \in \mathcal{X}}{\arg\max}\ acquisition(\mathbf{x}^*, w, t)$
    $y_t \leftarrow$ value from a respondent w.
    $\mathcal{D}_w \leftarrow \mathcal{D}_w \cup \{(x_t, y_t)\}$
    Recalculate $\mathbb{E}[y^*|\mathbf{x}^*, \mathcal{D}_w^{(t)}], \mathbb{V}[y^*|\mathbf{x}^*, \mathcal{D}_w^{(t)}]$
    Recalculate $\mathbf{r}_w^{(t)}$
  end for
  $\{\mathbf{r}_w\}_{w \in \mathcal{W}}, \{\boldsymbol{\mu}_k\}_{k \in [n_c]} = \underset{\{\mathbf{r}_w\}, \{\boldsymbol{\mu}_k\}}{\arg\min}\ \mathcal{L}$
end for

Experiments

We conducted experiments with real-world and synthesized data sets to answer our research question: whether CBO can optimize human preferences efficiently with less preferable interactions than Bayesian optimization without transfer learning. We also investigate the detailed behavior of CBO in terms of different hyper-parameter settings and different numbers of dimensions of the search space. The experiments were approved by the university's ethical review board. In all experiments, we used the Matern kernel ($\nu = 2.5$) for calculating the covariance matrix of the Gaussian process.

Data Sets

We collected preferences from the crowd for three domains (`Smartphone`, `Face`, `Voice`) with different statistics and prepared four other data sets with different dimensions consisting of synthesized personalized preference models. The preference models serve as the ground truth in the experiment.

Domains Table 2 summarizes their statistics.

- `Smartphone` is represented in two dimensions: 180 levels of "Hue" and 255 levels of "Saturation". All colors are represented in a search space of 45,900 possible values by multiplying these values.
- `Face` is created by changing the values of each of the four facial expression muscle parameters ("Eyebrow", "Eye", "Nose", and "Mouth") using FaceGen[1]. Since each parameter has 21 possible values, there are $21^4 = 194,481$ possible facial expression images.
- `Voice` contains Japanese voice clips, each of which is 10 seconds long and created by VOICEPEAK[2]. Their parameters are "Speed", "Pitch", and "Happiness Level". There are 76 levels of "Speed" and 151 levels of "Pitch" and "Happiness Level", resulting in 585,276 synthesized voices.
- `Synthesized` i are synthesized data sets with different numbers of dimensions (2,5,10) and the numbers of their possible values (11, 3). We sampled data from multivariate Gaussian distribution, and we assume there are three clusters of people, i.e., we assume three different Gaussian distributions, and each person belongs to only one of them.

Domain	#dimensions and #values	Size of Search Space
`Face`	4 (21, 21, 21, 21)	194,481
`Voice`	3 (76, 151, 151)	585,276
`Smartphone`	2 (180, 255)	45,900
`Synthesized 1`	2 (11 for each)	121
`Synthesized 2`	5 (11 for each)	161,051
`Synthesized 3`	5 (3 for each)	243
`Synthesized 4`	10 (3 for each)	59,049

Table 2: Data Set Statistics

Obtaining Personalized Preference Models For each of the first three domains, we recruited 300 people through AMT[3] (for `Smartphone` and `Face`) and Lancers[4] (for `Voice`) [5]. And for each w of the 300 people, we obtained personalized preference models $\mathcal{D}_w$ that represent the distri-

[1] https://facegen.com/
[2] https://www.ah-soft.com/voice/6nare/
[3] https://www.mturk.com/
[4] https://www.lancers.jp/
[5] We used Lancers for Voice because it supports the microtask template to incorporate voice data.

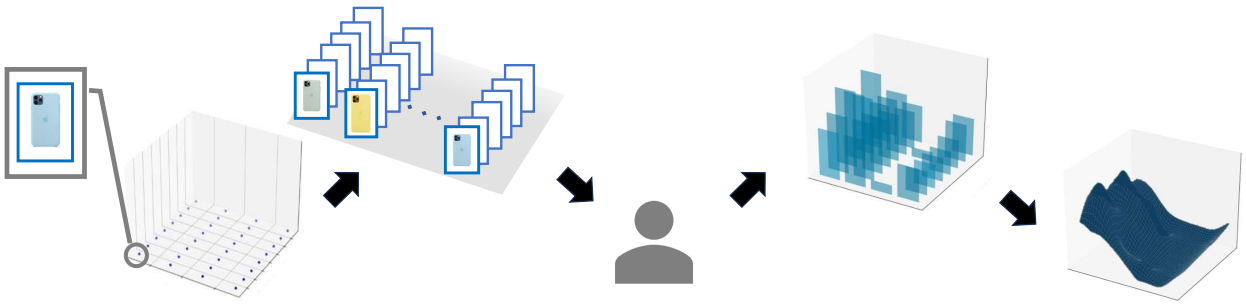


Figure 2: We generated $\mathcal{D}_w$ as follows: (1) We choose the representative points in the search space and generate the tasks to ask a crowd worker for his or her preference of each item (such as a smartphone) corresponding to each representative point (2) the Crowd worker give preference values for all representative points, and (3) we complemented the missing preference value in the search space.

bution of w's preference. Note that, we eliminated respondents who scored all items to the same score from the experiment.

Each $\mathcal{D}_w$ consists of preference values of w for every point in the search space (e.g., 45,900 points for `Smartphone`). Therefore, we generated $\mathcal{D}_w$ in the following way (Fig. 2).

1. We first divided the search space into the subspace in the same size (32, 54, 64 for `Smartphone`, `face`, and `Voice`. We divided the space based on the result of our preliminary experiment so that the number of values in each dimension is larger if we see a bigger change in preference in the dimension. For example, we chose four "Hues" and eight "Saturation" for Smartphones (Table 2) because we know that "Saturation" was more critical in our preliminary experiment.),
2. Then, we ask each crowd worker to give their preference (on a seven Likert scale) on the representative point of each subspace. For example, we asked them to give their preferences on $4 \times 8 = 32$ points for Smartphones. We paid 0.5 USD for each Smartphone task, considering the time we needed to answer the questions. We paid about 0.5 and 1 USD for Face and Voice tasks, respectively, with the same policy.
3. Finally, we applied the Gaussian process regression to the values of all pairs of the representative points to obtain the whole distribution. We normalized the function value from the respondents between 0 and 1.

Evaluation Metrics

We evaluate methods in the average of preference values for all respondents. Specifically, we evaluate the above criteria by *mean average preference value (MAPV)*. The definition of MAPV is as follows:

$$MAPV(t) = \frac{1}{|W|} \sum_{w \in W} y_w^{(t)} \tag{15}$$

where $y_w^{(t)}$ is a preference value of w in t-th query. The larger the minimum value of MAPV and the larger the final value, we define the overall process $\mathcal{D}$ as the better Bayesian optimization.

Preliminary Experiment

In the preliminary experiment, we verify how CBO is effective for achieving more preferable items while avoiding less-preferred items in simple parameter settings.

Procedure We applied CBO to `Smartphone` and compared the result with a baseline (GP-UCB). We set T, the number of queries for learning a preference model, to 20, because a large number of queries is impractical in real-world applications. The procedure is as follows.

1. Phase 1: We randomly chose 50 from the 300 respondents and asked each of them $T = 20$ queries chosen by GP-UCB. They answered the queries according to their preference models.
2. Phase 2: We used X-means to compute n_c (the number of clusters. See Table 1), and we randomly chose the remaining 250 respondents and iteratively applied CBO one by one to learn the preference model of each respondent. In the optimization process for each w of the 250 respondents, he or she returns answers to the queries according to his or her preference model $\mathcal{D}_w$.

In this preliminary experiment, hyper-parameter were as follows: $\alpha_1^{(t)} = \alpha_2^{(t)} = 1$ (GP-UCB had $\alpha_1^{(t)} = 1$). (We explore good hyper-parameters in the following experiment). To reduce the computational complexity of the experiment, we approximated the integral in eq. 8,9 by selecting the equally spaced reference points in the distribution.

Results Fig. 3 shows the results of the proposed method CBO and the baseline method GP-UCB. We repeated to apply the 20 queries seven times (The order of the respondents is shuffled after each iteration). The line represents the average and standard deviation of MAPV of the seven iterations. The result shows that CBO achieves a higher average MAPV than GP-UCB in all stages of the queries, which means our

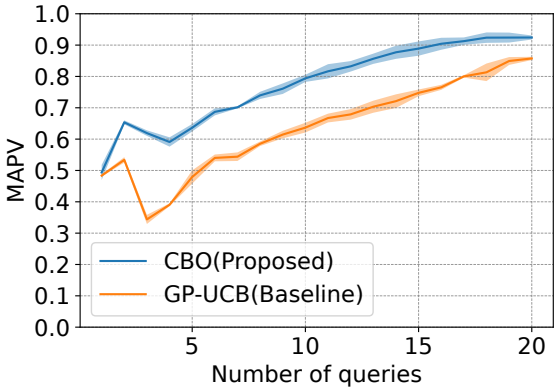


Figure 3: Result of Preliminary Experiment. When we assume simple parameter setting (CBO has $\alpha_1^{(t)} = \alpha_2^{(t)} = 1$, and GP-UCB has $\alpha_1^{(t)} = 1$), our method CBO outperforms the baseline.

CBO is more efficient in obtaining the respondents' preferences. Furthermore, we observed that GP-UCB obtained lower MAPV in the early stage of optimization because GP-UCB explores the items with low-preference values. In contrast, CBO achieves higher lowest MAPV. This indicates our transfer learning approach is beneficial for achieving higher MAPV faster and avoiding low-preferable items, even when the parameters are not well optimized.

Experiment 1: Effects of Hyper-Parameters

Given that CBO looks better than the baseline in terms of its efficiency and the number of shown unpreferable items, we explored good parameter settings with the three real-world data sets.

Procedure We repeated the procedure of the preliminary experiment with different parameter settings to find good setting strategies for the CBO hyper-parameters.

Since it is not realistic to exhaustively explore two independent parameters ($\alpha_1^{(t)}$ and $\alpha_2^{(t)}$) of the CBO acquisition function, we explored the optimal settings of $\alpha_1^{(t)}$ for GP-UCB (Experiment 1.1), and $\alpha_2^{(t)}$ of CBO when $\alpha_1^{(t)}$ is set to the optimal one found in Experiment 1.1 (Experiment 1.2).

We conducted this experiment with all three domains. We ran the experiment seven times for `Smartphone` and `Face` domain and five times for the `Voice`.

In experiment 1.1, we repeated the experiments with several parameter settings to determine whether it is better to control $\alpha_1^{(t)}$ to be monotonically increasing or decreasing, the timing to switch the $\alpha_1^{(t)}$ and E, and the initial value of $\alpha_1^{(t)}$. In experiment 1.2, we conducted the same process as experiment 1.1 with respect to $\alpha_2^{(t)}$.

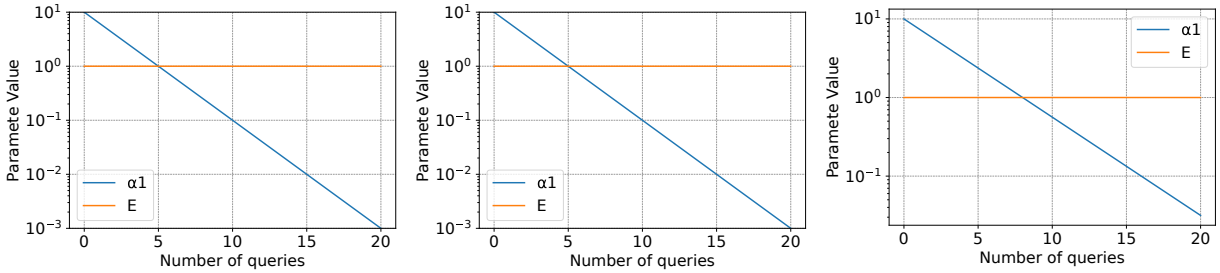
We set the change of $\alpha_1^{(t)}$ and $\alpha_2^{(t)}$ as $a \times 10^{(-b \times \frac{t}{20})}$. Therefore, by changing the values of a and b, we can control the behavior of each of the two parameters. In the equations, t represents that CBO is processing the t-th query, ranging from 1 to 20 in this experiment. The basic idea is that the importance of the expected value term, the standard deviation term, and the c-expected value term change as CBO processes new queries. The weight for the expected value term in the acquisition function is always 1, which is explicitly expressed as "E" to compare with the magnitude of $\alpha_1^{(t)}$ and $\alpha_2^{(t)}$.

Result of Experiment 1.1 Figure 4a show the good $\alpha_1^{(t)}$ settings and the orange lines in Figure 4c show the change of MAPV of GP-UCB. The figure on the left represents `Smartphone` domain, the middle represents `Face`, and the right represents `Voice`. In all domains, when setting $\alpha_1^{(t)}$ is monotonically decreasing, with an initial value of 10, GP-UCB achieves optimal MAPV. In this experiment, we note T_1 is the time that values of $\alpha_1^{(t)}$ and E are switched. We found there is a trade-off between the MAPV in the early stage of optimization and the final value of the MAPV, we assume the smallest T_1 such that the MAPV in the 20-th iteration exceeds a certain value p is the optimal T_1. In this experiment, p is set to 0.95, 0.75, and 0.9 for the `Smartphone`, `Face`, and `Voice` domains, respectively. As a result, in the three datasets, we found the optimal T_1 are $T_1 = 5$, $T_1 = 5$, and $T_1 = 8$, respectively.

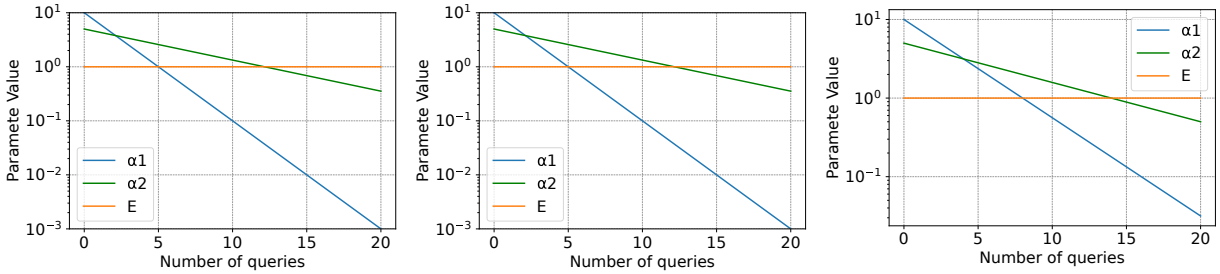
Result of Experiment 1.2 In the experiment where we compared the different $\alpha_2^{(t)}$ settings to each other while fixing $\alpha_1^{(t)}$ to 10 which was the best value obtained in Experiment 1.1, we found that the first value of $\alpha_2^{(t)}$ needs to be set less than $\alpha_1^{(t)}$ to show good performance. Actually, in all the three domains, we found that setting $\alpha_2^{(t)}$ monotonically decreasing with an initial value of 5, yields the optimal MAPV. Figure 4b show the best $\alpha_1^{(t)}$ and $\alpha_2^{(t)}$ settings in the experiment and the blue lines in Figure 4c show the change of MAPV of CBO.

Similar to Experiment 1.1, we note T_2 as the time that values of $\alpha_2^{(t)}$ and E are switched. For T_2, if T_2 increases to a certain value, there is no significant difference in the experimental results for the subsequent parameters. Figs 5 show the range of $\alpha_2^{(t)}$ that of MAPV is almost the same.

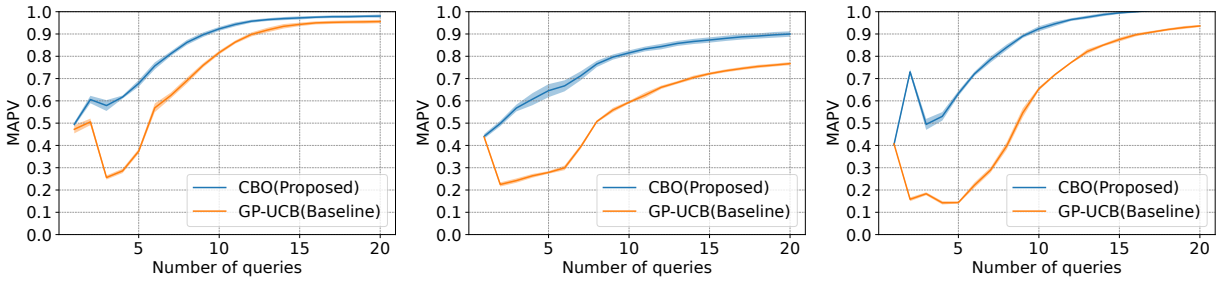
Discussion. The results show that the following hyper-parameter setting is good for CBO regardless of data sets: $\alpha_1^{(t)}$ is set to the largest in the beginning, then $\alpha_2^{(t)}$ becomes the largest, and finally, E becomes the largest. The setting can be explained as follows. In the first few queries during the exploration process, CBO's acquisition function sample points are expected to be a somewhat larger preference value by focusing on $\alpha_1^{(t)}$ while also taking into account information on the clusters. Then, by focusing on $\alpha_2^{(t)}$, points are samples for exploitation considering the c-expected value. Finally, by focusing on E, points at which the function values are expected to be large are sampled. This allows CBO to estimate the points respondents prefer with fewer iterations while avoiding interactions they do not prefer, compared to ordinary Bayesian optimization.



(a) Optimal settings of parameter $\alpha_1^{(t)}$ in each domain, which are used for GP-UCB. Note E indicates the weight for the expectation part in eq.7, which is constant 1.



(b) Optimal setting of parameters $\alpha_1^{(t)}$ and $\alpha_2^{(t)}$ which are used for CBO. Note, E indicates the weight for the expectation part in eq.13, which is constant 1.



(c) MAPV in each domain.

Figure 4: Results of Experiment 1.1 and 1.2 in each domain. Left: `Smartphone`, Center: `Face`, Right: `Voice`. (a)(b) show the optimal parameters by the process of Experiments 1.1 and 1.2. The lines in (c) show the results of each method using the optimal parameter settings. The results show our CBO achieves higher values in the earlier stage, and the lowest value in the optimization process is higher than GP-UCB. It indicates CBO is efficient in getting higher value while avoiding unpreferable items.

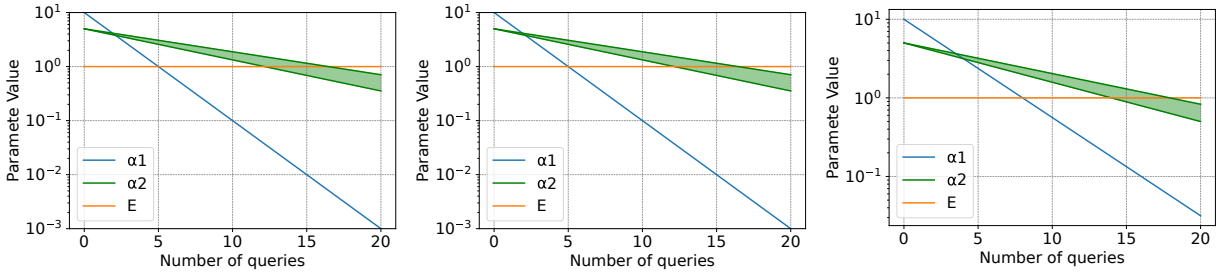
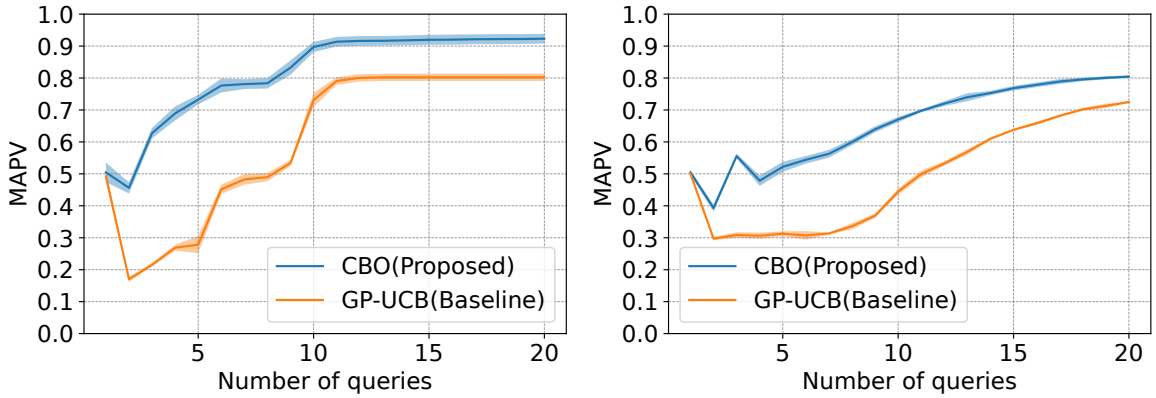
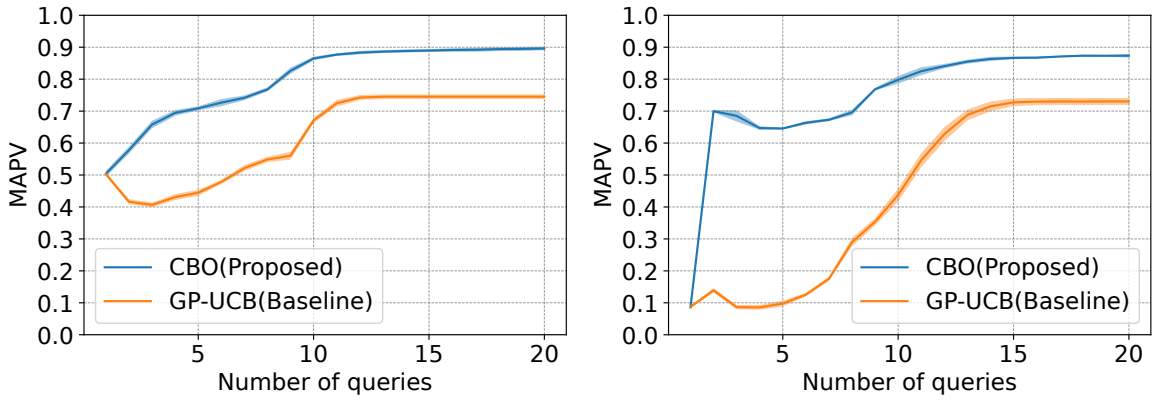


Figure 5: Valid range of $\alpha_2^{(t)}$ in each domain. Left: `Smartphone`, Center: `Face`, Right: `Voice`. In the area of α_2, CBO performs almost the same results in Experiment 1.2.



(a) MAPV in `Synthesized 1`. Dimension is 2; possible values are 11 for each dimension.

(b) MAPV in `Synthesized 2`. Dimension is 5; possible values are 11 for each dimension.



(c) MAPV in `Synthesized 3`. Dimension is 5; possible values are 3 for each dimension.

(d) MAPV in `Synthesized 4`. Dimension is 10; possible values are 3 for each dimension.

Figure 6: Result of experiments 2 using synthesized datasets. CBO is effective in every setting.

Experiment 2: Effects of Dimensions

Next, we explored the effects of the number of dimensions on the performance.

Procedure. We did the experiment in the same way as in Experiment 1, assuming that we fix the hyper-parameters to the best ones we found in the experiment, and we used the four Synthesized data sets `Synthesized 1`–`4` in Table 2.

Result. Figure 6 shows the results. It shows that CBO constantly achieves higher results than the baseline and is even more effective the high-dimensional settings.

Discussion. The result showed that CBO works further better compared to the baseline. This is not surprising because transfer learning changes the problem of inferring the preference model into the problem of finding an appropriate cluster the person belongs to.

Limitation

This work has several limitations. First, our experiment deals only with cases in which respondents answer their preferences accurately. When people respond to their preferences, they sometimes do not answer accurately and give noisy responses, but this study does not deal with such cases. We believe that we can introduce existing techniques to deal with noisy observations (Letham et al. 2019), but the application is out of the paper and interesting future work. Second, the maximum number of dimensions in our experiment is limited to ten. As we discussed in the previous section, CBO's transfer learning allows us to focus on choosing the best cluster if we already know the set of clusters. Note that in some cases, a larger number of dimensions may require a larger number of people to obtain clusters. However, this depends on the distribution and does not apply to all cases. Addressing the problem is independent of our problem, so we can combine our approach with some dimensionality reduction methods (Bouveyron and Brunet-Saumard 2014; Kriegel, Kröger, and Zimek 2009). Third, if the cluster to which a respondent belongs is incorrectly determined during the optimization process, the optimization may be less efficient than existing methods. We will address this issue by further improving the acquisition function in the future. Fourth, although cluster-aware transfer learning was successful in not only optimizing the personalized model efficiently and in reducing the number of unpreferred interac-

tions, it is an essential limitation that we cannot completely avoid such interactions.

Conclusion

In this study, we proposed a method named Cluster-aware Bayesian optimization (CBO), which considers the clusters of people to optimize efficiency. We conducted simulation experiments in three domains. We found that CBO is more efficient than the generic no-transfer Bayesian optimization method and that a good parameter setting strategy is to set the CBO hyper-parameters $\alpha_1^{(t)}$ and $\alpha_2^{(t)}$ to be the largest in the order of $\alpha_1^{(t)}$, $\alpha_2^{(t)}$, and E, respectively. The experiments show the strategy was effective for all three data sets with different solution dimensions and search space sizes.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Number JP21H03552, JP22H00508, JP22K17944, JST CREST Grant Number JPMJCR22M, Japan. We are grateful to Kaihotsu for helpful advice.

References

Adaji, I. 2017. Towards Improving E-Commerce Users Experience Using Personalization Persuasive Technology. In *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization (UMAP)*, 318–321.

Anshari, M.; Almunawar, M. N.; Lim, S. A.; and Al-Mudimigh, A. 2019. Customer relationship management and big data enabled: Personalization & customization of services. *Applied Computing and Informatics*, 15(2): 94–101.

Bardenet, R.; Brendel, M.; Kégl, B.; and Sebag, M. 2013. Collaborative hyperparameter tuning. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*, 199–207.

Bergstra, J.; and Bengio, Y. 2012. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(2).

Bouveyron, C.; and Brunet-Saumard, C. 2014. Model-based clustering of high-dimensional data: A review. *Computational Statistics & Data Analysis*, 71: 52–78.

Brochu, E. 2010. *Interactive Bayesian optimization: learning user preferences for graphics and animation*. Ph.D. thesis, University of British Columbia.

Brochu, E.; Cora, V. M.; and De Freitas, N. 2010. A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*.

Feurer, M.; Springenberg, J.; and Hutter, F. 2015. Initializing bayesian hyperparameter optimization via meta-learning. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 1128–1135.

Frazier, P. I. 2018. A tutorial on Bayesian optimization. *arXiv preprint arXiv:1807.02811*.

Gajos, K. Z.; Weld, D. S.; and Wobbrock, J. O. 2010. Automatically generating personalized user interfaces with Supple. *Artificial intelligence*, 174(12-13): 910–950.

Korzepa, M.; Petersen, M. K.; Larsen, J. E.; and Mørup, M. 2020. Simulation Environment for Guiding the Design of Contextual Personalization Systems in the Context of Hearing Aids. In *Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization (UMAP)*, 293–298. Association for Computing Machinery.

Kriegel, H.-P.; Kröger, P.; and Zimek, A. 2009. Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *ACM Transactions on Knowledge Discovery from Data*, 3(1): 1–58.

Letham, B.; Karrer, B.; Ottoni, G.; and Bakshy, E. 2019. Constrained Bayesian optimization with noisy experiments. *Bayesian Analysis*, 14(2): 495–519.

Liashchynskyi, P.; and Liashchynskyi, P. 2019. Grid search, random search, genetic algorithm: a big comparison for NAS. *arXiv preprint arXiv:1912.06059*.

Marco, A.; Berkenkamp, F.; Hennig, P.; Schoellig, A. P.; Krause, A.; Schaal, S.; and Trimpe, S. 2017. Virtual vs. real: Trading off simulations and physical experiments in reinforcement learning with Bayesian optimization. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 1557–1563.

Ohn-Bar, E.; Guerreiro, J.; Kitani, K.; and Asakawa, C. 2018. Variability in reactions to instructional guidance during smartphone-based assisted navigation of blind users. *Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies (IMWUT)*, 2(3): 1–25.

Papež, M.; and Quinn, A. 2022. Transferring model structure in Bayesian transfer learning for Gaussian process regression. *Knowledge-Based Systems*, 251: 108875.

Pelleg, D.; Moore, A. W.; et al. 2000. X-means: Extending k-means with efficient estimation of the number of clusters. In *Proceedings of International Conference on Machine Learning (ICML)*, 727–734.

Rasmussen, C. E.; and Williams, C. K. 2004. Gaussian processes in machine learning. *Lecture Notes in Computer Science*, 3176: 63–71.

Rentfrow, P. J.; and Gosling, S. D. 2003. The do re mi's of everyday life: the structure and personality correlates of music preferences. *Journal of personality and social psychology*, 84(6): 1236.

Schilling, N.; Wistuba, M.; and Schmidt-Thieme, L. 2016. Scalable hyperparameter optimization with products of gaussian process experts. In *Machine Learning and Knowledge Discovery in Databases: European Conference (ECML PKDD)*, 33–48. Springer.

Silva, N.; Carvalho, D.; Pereira, A. C.; Mourão, F.; and Rocha, L. 2019. The pure cold-start problem: A deep study about how to conquer first-time users in recommendations domains. *Information Systems*, 80: 1–12.

Snoek, J.; Larochelle, H.; and Adams, R. P. 2012. Practical bayesian optimization of machine learning algorithms. *Advances in Neural Information Processing Systems (NIPS)*, 1–9.

Sui, Y.; Gotovos, A.; Burdick, J.; and Krause, A. 2015. Safe exploration for optimization with Gaussian processes. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, 997–1005. PMLR.

Swersky, K.; Snoek, J.; and Adams, R. P. 2013. Multi-task bayesian optimization. *Advances in neural information processing systems (NIPS)*, 1–9.

Tyrväinen, O.; Karjaluoto, H.; and Saarijärvi, H. 2020. Personalization and hedonic motivation in creating customer experiences and loyalty in omnichannel retail. *Journal of Retailing and Consumer Services*, 57.

Wang, X.; Jin, Y.; Schmitt, S.; and Olhofer, M. 2022. Recent Advances in Bayesian Optimization. arXiv:2206.03301.

Wistuba, M.; Schilling, N.; and Schmidt-Thieme, L. 2018. Scalable gaussian process-based transfer surrogates for hyperparameter optimization. *Machine Learning*, 107(1): 43–78.

Yao, J.; Yang, L.; and Xu, X. 2022. Online Dependent Task Assignment in Preference Aware Spatial Crowdsourcing. *IEEE Transactions on Services Computing*, 1–4.