Active Learning for Distributionally Robust Level-Set Estimation

Yu Inatsu [1] **Shogo Iwazaki** [1] **Ichiro Takeuchi** [1 2]

Abstract

Many cases exist in which a black-box function f with high evaluation cost depends on two types of variables $\boldsymbol{x}$ and $\boldsymbol{w}$, where $\boldsymbol{x}$ is a controllable *design* variable and $\boldsymbol{w}$ are uncontrollable *environmental* variables that have random variation following a certain distribution P. In such cases, an important task is to find the range of design variables $\boldsymbol{x}$ such that the function $f(\boldsymbol{x}, \boldsymbol{w})$ has the desired properties by incorporating the random variation of the environmental variables $\boldsymbol{w}$. A natural measure of robustness is the probability that $f(\boldsymbol{x}, \boldsymbol{w})$ exceeds a given threshold h, which is known as the *probability threshold robustness* (PTR) measure in the literature on robust optimization. However, this robustness measure cannot be correctly evaluated when the distribution P is unknown. In this study, we addressed this problem by considering the *distributionally robust PTR* (DRPTR) measure, which considers the worst-case PTR within given candidate distributions. Specifically, we studied the problem of efficiently identifying a reliable set H, which is defined as a region in which the DRPTR measure exceeds a certain desired probability α, which can be interpreted as a level set estimation (LSE) problem for DRPTR. We propose a theoretically grounded and computationally efficient active learning method for this problem. We show that the proposed method has theoretical guarantees on convergence and accuracy, and confirmed through numerical experiments that the proposed method outperforms existing methods.

1. Introduction

In the manufacturing industry, product performance often depends on two types of variables: design variables and environmental variables. The design variables are completely controllable, whereas environmental variables are random variables that change depending on the usage environment of the product. When considering such a problem, it is important to identify the design variables that allow the product performance to exceed the desired requirement threshold with a sufficiently high degree of confidence, taking into account the randomness of the environmental variables. In this setting, we must emphasize that there are two distinctly different phases of the product: the development phase and the use phase. In the development phase, we have full control over the design variables and environmental variables. In the use phase, on the other hand, the design variables are fixed, and the environmental variables change randomly and cannot be controlled.

Let $f(\boldsymbol{x}, \boldsymbol{w})$ represent the performance of the product, and let $h \in \mathbb{R}$ be a desired performance threshold, where $\boldsymbol{x}$ is a design variable defined on $\mathcal{X}$, and $\boldsymbol{w}$ is an environmental variable defined on Ω. Then, we consider the following robustness measure:

$$\mathrm{PTR}(\boldsymbol{x}) = \int_{\Omega} \mathbb{1}[f(\boldsymbol{x}, \boldsymbol{w}) > h] p^{\dagger}(\boldsymbol{w}) \mathrm{d}\boldsymbol{w},$$

where $\mathbb{1}[\cdot]$ is the indicator function and $p^{\dagger}(\boldsymbol{w})$ is the probability density function of $\boldsymbol{w}$. This measure is called the probability threshold robustness (PTR) measure in the field of robust optimization (Beyer & Sendhoff, 2007), and can be interpreted as a measure of how well the design variables behave under randomness in the environmental variables. In the manufacturing industry, it is desirable to identify the set of controllable variables $\boldsymbol{x} \in \mathcal{X}$ for which $\mathrm{PTR}(\boldsymbol{x})$ is greater than a certain threshold. In other words, this problem is interpreted as a level-set estimation (LSE) (Bryan et al., 2006; Gotovos et al., 2013) of the PTR measure. There are two main reasons for considering LSE of the PTR measure. One is that by enumerating all the design variables that exceed the desired threshold with a high probability, it is possible to respond the usage conditions of various users. The other is to consider some optimization problem (e.g., to find $\boldsymbol{x}$ with the minimum price) for design variables with PTR measures above a certain level. This is known as the chance-constrained programming problem (Charnes & Cooper, 1959), and has many applications such as finance, in addition to manufacturing industry. Unfortunately, however, the PTR measure cannot be correctly evaluated when

[1] Department of Computer Science, Nagoya Institute of Technology, Aichi, Japan [2] RIKEN Center for Advanced Intelligence Project, Tokyo, Japan. Correspondence to: Ichiro Takeuchi <takeuchi.ichiro@nitech.ac.jp>.

Proceedings of the 38^{th} *International Conference on Machine Learning*, PMLR 139, 2021. Copyright 2021 by the author(s).

$p^\dagger(\boldsymbol{w})$ is unknown. If $p^\dagger(\boldsymbol{w})$ is unknown and the estimated density is simply plugged in, then $\mathrm{PTR}(\boldsymbol{x})$ is no longer valid as a robustness measure because of the estimation error.

In this study, we considered a distributionally robust PTR (DRPTR) measure, which includes uncertainty about $p^\dagger(\boldsymbol{w})$ under the setting that $p^\dagger(\boldsymbol{w})$ is unknown. Let $\mathcal{A}$ be a user-specified class of candidate distributions of $\boldsymbol{w}$. Then, the DRPTR measure can be defined as

$$F(\boldsymbol{x}) = \inf_{p(\boldsymbol{w}) \in \mathcal{A}} \int_\Omega \mathbb{1}[f(\boldsymbol{x}, \boldsymbol{w}) > h] p(\boldsymbol{w}) \mathrm{d}\boldsymbol{w}.$$

The DRPTR measure has the advantage of being robust with respect to using wrong distributions because it can be interpreted as the PTR in the worst case among the candidate distributions. In this study, we formulated this problem as an active learning problem for the LSE for $F(\boldsymbol{x})$ instead of $\mathrm{PTR}(\boldsymbol{x})$, and developed a theoretically grounded and numerically efficient algorithm for its calculation. Figure 1 shows the conceptual diagram for this study. The basic ideas of our proposed method are as follows. First, we consider the function $f(\boldsymbol{x}, \boldsymbol{w})$ to be a black-box function with a high evaluation cost, and we employ a Gaussian process (GP) model as a surrogate model. Next, we predict the target DRPTR measure using the GP model for the black-box function $f(\boldsymbol{x}, \boldsymbol{w})$. Finally, we perform LSE using credible intervals of the DRPTR measure calculated on the basis of this prediction.

1.1. Related Work

Active learning using GP models (Williams & Rasmussen, 2006) for black-box functions have been actively studied in the context of Bayesian optimization (see, e.g., (Settles, 2009; Shahriari et al., 2016)). Several studies have been conducted on active learning for LSE (Bryan et al., 2006; Gotovos et al., 2013; Zanette et al., 2018; Inatsu et al., 2020a). Furthermore, some researchers applied LSE to efficiently identify safety regions (Sui et al., 2015; Turchetta et al., 2016; Sui et al., 2018; Wachi et al., 2018), and others used LSE to enumerate the local minima of black-box functions (Inatsu et al., 2020c).

Many studies have been conducted on active learning under input uncertainty (including random environmental variables). In (Inatsu et al., 2020b), the authors proposed an efficient method for performing LSE in the setting where the input is a random variable generated from a certain distribution. In other studies, the researchers formulated the randomness of the input with some robustness measures for performing active learning on it. For example, the authors of (Bogunovic et al., 2018) used the worst-case function value of the input shift as a robustness measure. Similarly, other research ((Beland & Nair, 2017; Toscano-Palmerin & Frazier, 2018; Oliveira et al., 2019; Fröhlich et al., 2020; Gessner et al., 2020; Iwazaki et al., 2021)) dealt with the stochastic robustness (SR) measure, which is a robustness measure defined by integrating the black-box function against the input distribution. In another study closely related to the present work, the authors of (Iwazaki et al., 2020a) proposed an active learning method for LSE in the PTR measure on the basis of random inputs; in (Iwazaki et al., 2021), the authors considered an active learning method for both LSE and maximization problems in the PTR measure. However, these two are not distributionally robust settings. Distributionally robust optimization (DRO), which is not an active learning framework, was first introduced by (Scarf, 1958). DRO is an important topic in the context of robust optimization, and there have been countless related studies (see (Rahimian & Mehrotra, 2019) for comprehensive survey of DRO). Active learning methods for DRO with uncertainty environmental variables have recently been proposed by (Kirschner et al., 2020; Nguyen et al., 2020). The main differences to our problem setup are that they focus on a distributionally robust SR (DRSR) measure for the target function, which is the worst-case SR measure in candidate distributions of the unknown environmental variable, and consider the maximization problem for the DRSR measure. In particular, for the former, we cannot directly apply their proposed methods and theoretical techniques because the target function is different from ours. To the best of our knowledge, none of these studies have addressed the same research problem considered in the present work.

1.2. Contributions

The main contributions of this study are summarized as follows:

- We formulate the LSE problem for the DRPTR measure, i.e., the problem of finding the set of design variables for which the DRPTR measure exceeds a given threshold.

- We construct non-trivial credible intervals for the DRPTR measure and propose a new acquisition function (AF) based on an expected classification improvement. Using them, we propose an active learning method for the LSE of the DRPTR measure. Moreover, because the naive implementation of our proposed AF requires a large computational cost, we propose a computationally efficient technique for its calculation.

- We clarify the theoretical property of the proposed method. Under mild conditions, we show that the proposed method has desirable accuracy and convergence properties.

- We describe the empirical performance of the proposed method through the results of numerical experiments with benchmark functions and infection simulations.

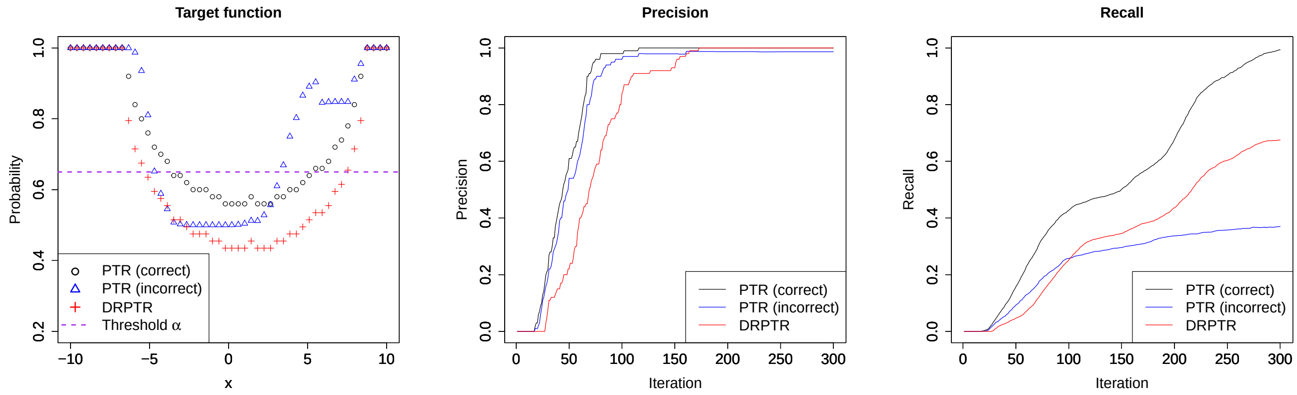


Figure 1. An example of our problem setup. In the left figure, black circles and blue triangles represent values of PTR by using the correct and incorrect distribution of the environmental variable $\boldsymbol{w}$, respectively. Also, red crosses represent values of DRPTR by using the correct distribution of $\boldsymbol{w}$ as the reference distribution. It can be seen that DRPTR is smaller than PTR (correct) because DRPTR is defined by taking the inf operation. This implies that DRPTR is more conservative than PTR (incorrect) with respect to the precision for PTR (correct). The central and right figures represent the precision and recall, respectively. From the central figure, the precision of PTR (incorrect) does not reach 1, while DRPTR reaches 1. Therefore, DRPTR is superior in the sense that it can be more conservative than PTR (incorrect), depending on the choice of reference distribution. Our goal is to efficiently find all design variables where DRPTR is greater than a given threshold α (purple dashed line).

2. Preliminary

Let $f : \mathcal{X} \times \Omega \to \mathbb{R}$ be an expensive-to-evaluate black-box function. We assume that $\mathcal{X}$ and Ω are finite sets. For each input $(\boldsymbol{x}, \boldsymbol{w}) \in \mathcal{X} \times \Omega$, the value of $f(\boldsymbol{x}, \boldsymbol{w})$ is observed as $y = f(\boldsymbol{x}, \boldsymbol{w}) + \varepsilon$ with an independent noise ε, where ε follows Gaussian distribution $\mathcal{N}(0, \sigma^2)$. In our setting, a variable $\boldsymbol{w} \in \Omega$ stochastically fluctuates by the (unknown) discrete distribution $P^\dagger$ in the use phase, whereas we can specify $\boldsymbol{w}$ in the development phase. Moreover, let $\mathcal{A}$ be a family of candidate distributions of $P^\dagger$. In this work, we consider $\mathcal{A} = \{\text{p.m.f. } p(\boldsymbol{w}) \mid d(p(\boldsymbol{w}), p^*(\boldsymbol{w})) < \epsilon\}$, where $p^*(\boldsymbol{w})$ is a user-specified reference distribution, $d(\cdot, \cdot)$ is a given distance metric between two distributions, and $\epsilon > 0$. Here, we assume that $p^*(\boldsymbol{w})$ is specified by users. For example, if there is no prior information about $P^\dagger$, one way is to use the uniform distribution as $p^*(\boldsymbol{w})$, or if the empirical distribution of $P^\dagger$ is available, a reasonable choice is to use it as $p^*(\boldsymbol{w})$. The estimation of $p^*(\boldsymbol{w})$ is out of scope of this paper. Then, under the given threshold h, we define the DRPTR $F(\boldsymbol{x})$ for each $\boldsymbol{x} \in \mathcal{X}$ as

$$F(\boldsymbol{x}) = \inf_{p(\boldsymbol{w}) \in \mathcal{A}} \sum_{\boldsymbol{w} \in \Omega} \mathbb{1}[f(\boldsymbol{x}, \boldsymbol{w}) > h] p(\boldsymbol{w}).$$

The aim of this study was to efficiently identify a subset H of $\mathcal{X}$ that satisfies $F(\boldsymbol{x}) > \alpha$ for a given threshold $\alpha \in (0, 1)$:

$$H = \{\boldsymbol{x} \in \mathcal{X} \mid F(\boldsymbol{x}) > \alpha\}. \tag{2.1}$$

Moreover, we define the lower set L as $L = \{\boldsymbol{x} \in \mathcal{X} \mid F(\boldsymbol{x}) \leq \alpha\}$.

Gaussian Process In this study, we used the Gaussian process (GP) to model the unknown black-box function f. First, we assume that the GP, $\mathcal{GP}(0, k((\boldsymbol{x}, \boldsymbol{w}), (\boldsymbol{x}', \boldsymbol{w}')))$ is a prior distribution of f, where $k((\boldsymbol{x}, \boldsymbol{w}), (\boldsymbol{x}', \boldsymbol{w}'))$ is a positive-definite kernel. Then, given the dataset $\{(\boldsymbol{x}_i, \boldsymbol{w}_i, y_i)\}_{i=1}^t$, the posterior distribution of f also follows the GP, and its posterior mean $\mu_t(\boldsymbol{x}, \boldsymbol{w})$ and posterior variance $\sigma_t^2(\boldsymbol{x}, \boldsymbol{w})$ are given by

$$\begin{aligned} \mu_t(\boldsymbol{x}, \boldsymbol{w}) &= \boldsymbol{k}_t^\top(\boldsymbol{x}, \boldsymbol{w})(\boldsymbol{K}_t + \sigma^2 \boldsymbol{I}_t)^{-1} \boldsymbol{y}_t, \\ \sigma_t^2(\boldsymbol{x}, \boldsymbol{w}) &= k((\boldsymbol{x}, \boldsymbol{w}), (\boldsymbol{x}, \boldsymbol{w})) \\ &\qquad - \boldsymbol{k}_t^\top(\boldsymbol{x}, \boldsymbol{w})(\boldsymbol{K}_t + \sigma^2 \boldsymbol{I}_t)^{-1} \boldsymbol{k}_t(\boldsymbol{x}, \boldsymbol{w}), \end{aligned}$$

where $\boldsymbol{k}_t(\boldsymbol{x}, \boldsymbol{w})$ is the t-dimensional vector whose jth element is $k((\boldsymbol{x}, \boldsymbol{w}), (\boldsymbol{x}_j, \boldsymbol{w}_j))$, $\boldsymbol{y}_t = (y_1, \ldots, y_t)^\top$, $\boldsymbol{I}_t$ is the $t \times t$ identity matrix, and $\boldsymbol{K}_t$ is the $t \times t$ matrix whose (j, k)th element is $k((\boldsymbol{x}_j, \boldsymbol{w}_j), (\boldsymbol{x}_k, \boldsymbol{w}_k))$.

3. Proposed Method

In this section, we propose an active learning method for efficiently identifying (2.1). The target function $F(\boldsymbol{x})$ is a random variable because $F(\boldsymbol{x})$ is the function of $f(\boldsymbol{x}, \boldsymbol{w})$, and $f(\boldsymbol{x}, \boldsymbol{w})$ is drawn from GP. Thus, a reasonable method to identify (2.1) is to construct a credible interval of $F(\boldsymbol{x})$, and estimate H using the lower bound of the constructed credible interval. Unfortunately, although $f(\boldsymbol{x}, \boldsymbol{w})$ follows GP, $F(\boldsymbol{x})$ does not follow GP. Hence, the credible interval of $F(\boldsymbol{x})$ cannot be directly calculated on the basis of normal distributions. In the next section, we propose a simple and theoretically valid credible interval of $F(\boldsymbol{x})$ using the credible interval of $f(\boldsymbol{x}, \boldsymbol{w})$.

3.1. Credible Interval and LSE

For any input $(\boldsymbol{x}, \boldsymbol{w}) \in \mathcal{X} \times \Omega$ and step t, we define a credible interval of $f(\boldsymbol{x}, \boldsymbol{w})$ as $Q_t(\boldsymbol{x}, \boldsymbol{w}) = [l_t(\boldsymbol{x}, \boldsymbol{w}), u_t(\boldsymbol{x}, \boldsymbol{w})]$, where $l_t(\boldsymbol{x}, \boldsymbol{w}) = \mu_t(\boldsymbol{x}, \boldsymbol{w}) - \beta_t^{1/2}\sigma_t(\boldsymbol{x}, \boldsymbol{w})$, $u_t(\boldsymbol{x}, \boldsymbol{w}) = \mu_t(\boldsymbol{x}, \boldsymbol{w}) + \beta_t^{1/2}\sigma_t(\boldsymbol{x}, \boldsymbol{w})$, and $\beta_t^{1/2} \geq 0$. Similarly, we define a credible interval of $\mathbb{1}[f(\boldsymbol{x}, \boldsymbol{w}) > h]$ on the basis of $Q_t(\boldsymbol{x}, \boldsymbol{w})$. For the theoretical analysis described in Section 4, we introduce a user-specified accuracy parameter $\eta > 0$. Specifically, we define the credible interval of $\mathbb{1}[f(\boldsymbol{x}, \boldsymbol{x}) > h]$ at step t as

$$\begin{aligned}
&\tilde{Q}_t(\boldsymbol{x}, \boldsymbol{w}; \eta) \equiv [\tilde{l}_t(\boldsymbol{x}, \boldsymbol{w}; \eta), \tilde{u}_t(\boldsymbol{x}, \boldsymbol{w}; \eta)] \\
&= \begin{cases} [1,1] & \text{if } l_t(\boldsymbol{x}, \boldsymbol{w}) > h - \eta, \\ [0,1] & \text{if } l_t(\boldsymbol{x}, \boldsymbol{w}) \leq h - \eta \text{ and } u_t(\boldsymbol{x}, \boldsymbol{w}) > h, \\ [0,0] & \text{if } l_t(\boldsymbol{x}, \boldsymbol{w}) \leq h - \eta \text{ and } u_t(\boldsymbol{x}, \boldsymbol{w}) \leq h. \end{cases}
\end{aligned}$$

Note that when the accuracy parameter $\eta = 0$, this credible interval simply indicates that if the lower (resp. upper) bound of $f(\boldsymbol{x}, \boldsymbol{w})$ is greater (resp. smaller) than h, we say that $\mathbb{1}[f(\boldsymbol{x}, \boldsymbol{w}) > h] = 1$ (resp. 0). Thus, a credible interval $Q_t^{(F)}(\boldsymbol{x}; \eta) \equiv [l_t^{(F)}(\boldsymbol{x}; \eta), u_t^{(F)}(\boldsymbol{x}; \eta)]$ of the target function $F(\boldsymbol{x})$ can be given by

$$\begin{aligned}
l_t^{(F)}(\boldsymbol{x}; \eta) &= \inf_{p(\boldsymbol{w}) \in \mathcal{A}} \sum_{\boldsymbol{w} \in \Omega} \tilde{l}_t(\boldsymbol{x}, \boldsymbol{w}; \eta) p(\boldsymbol{w}), \\
u_t^{(F)}(\boldsymbol{x}; \eta) &= \inf_{p(\boldsymbol{w}) \in \mathcal{A}} \sum_{\boldsymbol{w} \in \Omega} \tilde{u}_t(\boldsymbol{x}, \boldsymbol{w}; \eta) p(\boldsymbol{w}).
\end{aligned} \tag{3.1}$$

Note that if we use the $L1$ (or $L2$)-norm as the distance function $d(\cdot, \cdot)$, equation (3.1) is equivalent to solving a linear (or second-order cone) programming problem. In both cases, because solvers exist that can compute the optimal solution quickly, it is easy to compute $Q_t^{(F)}(\boldsymbol{x}; \eta)$ when using such distance functions. Then, we estimate H and L using $Q_t^{(F)}(\boldsymbol{x}; \eta)$ as follows:

$$\begin{aligned}
H_t &= \{\boldsymbol{x} \in \mathcal{X} \mid l_t^{(F)}(\boldsymbol{x}; \eta) > \alpha\}, \\
L_t &= \{\boldsymbol{x} \in \mathcal{X} \mid u_t^{(F)}(\boldsymbol{x}; \eta) \leq \alpha\}.
\end{aligned}$$

Also, we define the unclassified set as $U_t = \mathcal{X} \setminus (H_t \cup L_t)$.

3.2. Acquisition Function

In this section, we propose two acquisition functions to select the next evaluation point. Our proposed acquisition functions are based on the maximum improvement in level-set estimation (MILE) strategy proposed in (Zanette et al., 2018). In MILE, the expected value of the increase in the number of classifications after adding the new point $(\boldsymbol{x}^*, \boldsymbol{w}^*)$ is calculated, and the point with the largest expected value is selected. In this study, owing to the computational cost of calculating the acquisition function, we consider a strategy based on the expected value where points in the unclassified set are classified as H.

Let $(\boldsymbol{x}^*, \boldsymbol{w}^*)$ be a new point, and let $y^* = f(\boldsymbol{x}^*, \boldsymbol{w}^*) + \varepsilon$ be a new observation at point $(\boldsymbol{x}^*, \boldsymbol{w}^*)$. Furthermore, let $l_t^{(F)}(\boldsymbol{x}; 0 | \boldsymbol{x}^*, \boldsymbol{w}^*, y^*)$ be the lower bound of the credible interval of $F(\boldsymbol{x})$, where $\eta = 0$ when $(\boldsymbol{x}^*, \boldsymbol{w}^*, y^*)$ is newly added. Then, we consider the function $a_t(\boldsymbol{x}^*, \boldsymbol{w}^*)$:

$$a_t(\boldsymbol{x}^*, \boldsymbol{w}^*) = \sum_{\boldsymbol{x} \in U_t} \mathbb{E}_{y^*}[\mathbb{1}[l_t^{(F)}(\boldsymbol{x}; 0 | \boldsymbol{x}^*, \boldsymbol{w}^*, y^*) > \alpha]]. \tag{3.2}$$

In this work, we do not directly use (3.2) as the acquisition function because the value of (3.2) is sometimes exactly zero for any point. A reasonable method to avoid this problem is to consider a different function $b_t(\boldsymbol{x}^*, \boldsymbol{w}^*)$ only when the values of (3.2) are all zero. For theoretical treatment, we follow the strategy described in (Zanette et al., 2018), and consider the acquisition function of the form $\max\{a_t(\boldsymbol{x}^*, \boldsymbol{w}^*), \gamma b_t(\boldsymbol{x}^*, \boldsymbol{w}^*)\}$ with a positive constant parameter γ. Note that if we use a sufficiently small γ, it is almost the same when considering $b_t(\boldsymbol{x}^*, \boldsymbol{w}^*)$ only when the values of (3.2) are all zero; otherwise, $a_t(\boldsymbol{x}^*, \boldsymbol{w}^*)$. In Section 4, we present the theoretical guarantees of our proposed method for this acquisition function. In this section, we propose two types of $b_t(\boldsymbol{x}^*, \boldsymbol{w}^*)$. The first is based on the RMILE acquisition function proposed by (Zanette et al., 2018). The basic idea of RMILE is to add an additional variance term $\gamma\sigma_t(\boldsymbol{x}^*, \boldsymbol{w}^*)$ to the original MILE acquisition function. By using the same argument, we define the following modified acquisition function:

Definition 3.1 (Proposed acquisition function 1)**.** Let $a_t(\boldsymbol{x}^*, \boldsymbol{w}^*)$ be the function defined by (3.2), and let γ be a positive parameter. Then, we propose the following acquisition function $a_t^{(1)}(\boldsymbol{x}^*, \boldsymbol{w}^*)$:

$$a_t^{(1)}(\boldsymbol{x}^*, \boldsymbol{w}^*) = \max\{a_t(\boldsymbol{x}^*, \boldsymbol{w}^*), \gamma\sigma_t(\boldsymbol{x}^*, \boldsymbol{w}^*)\}.$$

Moreover, we select the next evaluation point $(\boldsymbol{x}_{t+1}, \boldsymbol{w}_{t+1})$ by maximizing $a_t^{(1)}(\boldsymbol{x}^*, \boldsymbol{w}^*)$.

The other acquisition function we propose uses $\gamma\text{RMILE}_t(\boldsymbol{x}^*, \boldsymbol{w}^*)$ instead of $\gamma\sigma_t(\boldsymbol{x}^*, \boldsymbol{w}^*)$ as the function $b_t(\boldsymbol{x}^*, \boldsymbol{w}^*)$, where $\text{RMILE}_t(\boldsymbol{x}^*, \boldsymbol{w}^*)$ is the RMILE function proposed in (Zanette et al., 2018).

Definition 3.2 (Proposed acquisition function 2)**.** Let $a_t(\boldsymbol{x}^*, \boldsymbol{w}^*)$ be the function defined by (3.2), and let γ be a positive parameter. Then, we propose the following acquisition function $a_t^{(2)}(\boldsymbol{x}^*, \boldsymbol{w}^*)$:

$$a_t^{(2)}((\boldsymbol{x}^*, \boldsymbol{w}^*)) = \max\{a_t(\boldsymbol{x}^*, \boldsymbol{w}^*), \gamma\text{RMILE}_t(\boldsymbol{x}^*, \boldsymbol{w}^*)\}.$$

Moreover, we select the next evaluation point $(\boldsymbol{x}_{t+1}, \boldsymbol{w}_{t+1})$ by maximizing $a_t^{(2)}(\boldsymbol{x}^*, \boldsymbol{w}^*)$.

Algorithm 1 Active learning for distributionally robust level-set estimation

```
Require: GP prior $\mathcal{GP}(0,\ k)$, threshold $h \in \mathbb{R}$, probability $\alpha \in (0, 1)$, accuracy parameter $\eta > 0$, tradeoff parameter $\{\beta_t\}_{t \leq T}$
  $H_0 \leftarrow \emptyset$, $L_0 \leftarrow \emptyset$, $U_0 \leftarrow \mathcal{X}$, $t \leftarrow 1$
  while $U_{t-1} \neq \emptyset$ do
    Compute $l_t^{(F)}(\boldsymbol{x}; \eta)$ and $u_t^{(F)}(\boldsymbol{x}; \eta)$ for all $\boldsymbol{x} \in \mathcal{X}$
    Choose $(\boldsymbol{x}_t, \boldsymbol{w}_t)$ by $(\boldsymbol{x}_t, \boldsymbol{w}_t) = \mathrm{argmax}_{(\boldsymbol{x}^*, \boldsymbol{w}^*) \in \mathcal{X} \times \Omega}\, a_{t-1}^{(1)}(\boldsymbol{x}^*, \boldsymbol{w}^*)$ (or $a_{t-1}^{(2)}(\boldsymbol{x}^*, \boldsymbol{w}^*)$ instead of $a_{t-1}^{(1)}(\boldsymbol{x}^*, \boldsymbol{w}^*)$)
    Observe $y_t \leftarrow f(\boldsymbol{x}_t, \boldsymbol{w}_t) + \varepsilon_t$
    Update GP by adding $((\boldsymbol{x}_t, \boldsymbol{w}_t), y_t)$ and compute $H_t, L_t$ and $U_t$
    $t \leftarrow t + 1$
  end while
  $\hat{H} \leftarrow H_{t-1}, \hat{L} \leftarrow L_{t-1}$
Ensure: Estimated Set $\hat{H}, \hat{L}$
```

The pseudocode of the proposed method is given in Algorithm 1.

3.3. Computational Techniques

Our proposed acquisition functions are based on (3.2), where (3.2) includes the calculation of the expected value. This expectation cannot be expressed as a simple expression using the cumulative distribution function (CDF) of the standard normal distribution, as in the original MILE (Zanette et al., 2018). One way to solve this problem is to generate many samples from the posterior distribution of y^* and numerically calculate the expected value. However, because one optimization calculation is required to calculate $\mathbb{1}[l_t^{(F)}(\boldsymbol{x}; 0 | \boldsymbol{x}^*, \boldsymbol{w}^*, y^*) > \alpha]$, if the expected value is calculated using M samples, then $M|U_t|$ optimization calculations are required to calculate $a_t(\boldsymbol{x}^*, \boldsymbol{w}^*)$ for each $(\boldsymbol{x}^*, \boldsymbol{w}^*)$. Therefore, to calculate $a_t(\boldsymbol{x}^*, \boldsymbol{w}^*)$ for all candidate points, $M|U_t||\mathcal{X} \times \Omega|$ optimization calculations are required. To reduce this large computational cost, we provide useful lemmas for efficiently computing the acquisition function. The expected values in (3.2) can be exactly calculated using the following lemma:

Lemma 3.1. Let $l_t(\boldsymbol{x}, \boldsymbol{w}_j | \boldsymbol{x}^*, \boldsymbol{w}^*, y^*)$ be the lower confidence bound of $f(\boldsymbol{x}, \boldsymbol{w}_j)$ after adding $(\boldsymbol{x}^*, \boldsymbol{w}^*, y^*)$ to $\{(\boldsymbol{x}_i, \boldsymbol{w}_i, y_i)\}_{i=1}^t$. Furthermore, let r_j be a number satisfying $h = l_t(\boldsymbol{x}, \boldsymbol{w}_j | \boldsymbol{x}^*, \boldsymbol{w}^*, r_j)$, and let $r^{(j)}$ be the jth-smallest number in the range r_1 to $r_{|\Omega|}$. For each $s \in \{1, \ldots, |\Omega| + 1\} \equiv [|\Omega| + 1]$, define $R_s = (r^{(s-1)}, r^{(s)}]$, where $r^{(0)} = -\infty$ and $r^{(|\Omega|+1)} = \infty$. Moreover, let c_s be a real number satisfying $c_s \in R_s$. Then, $\mathbb{E}_{y^*}[\mathbb{1}[l_t^{(F)}(\boldsymbol{x}; 0 | \boldsymbol{x}^*, \boldsymbol{w}^*, y^*) > \alpha]]$ can be calculated as follows:

$$
\begin{aligned}
&\mathbb{E}_{y^*}[\mathbb{1}[l_t^{(F)}(\boldsymbol{x}; 0 | \boldsymbol{x}^*, \boldsymbol{w}^*, y^*) > \alpha]] \\
&= \sum_{s=1}^{|\Omega|+1} \mathbb{P}(y^* \in R_s) \mathbb{1}[l_t^{(F)}(\boldsymbol{x}; 0 | \boldsymbol{x}^*, \boldsymbol{w}^*, c_s) > \alpha].
\end{aligned}
\tag{3.3}
$$

Lemma 3.1 implies that $|\Omega| + 1$ optimization calculations are required to calculate $\mathbb{E}_{y^*}[\mathbb{1}[l_t^{(F)}(\boldsymbol{x}; 0 | \boldsymbol{x}^*, \boldsymbol{w}^*, y^*) > \alpha]]$, but the following lemma shows that the number of optimization calculations can be reduced by checking a simple inequality:

Lemma 3.2. Let $c_1, \ldots, c_{|\Omega|+1}$ be numbers defined as in Lemma 3.1. Suppose that c_s satisfies

$$
\sum_{\boldsymbol{w} \in \Omega} \mathbb{1}[l_t(\boldsymbol{x}, \boldsymbol{w} | \boldsymbol{x}^*, \boldsymbol{w}^*, c_s) > h] p^*(\boldsymbol{w}) \leq \alpha.
$$

Then, $\mathbb{1}[l_t^{(F)}(\boldsymbol{x}; 0 | \boldsymbol{x}^*, \boldsymbol{w}^*, c_s) > \alpha] = 0$.

Finally, noting that $0 \leq \mathbb{P}(y^* \in R_s) \leq 1$ and $0 \leq \mathbb{1}[l_t^{(F)}(\boldsymbol{x}; 0 | \boldsymbol{x}^*, \boldsymbol{w}^*, c_s) > \alpha] \leq 1$, we can approximate (3.3) with any approximation accuracy $\zeta > 0$:

Lemma 3.3. Let $\zeta > 0$, and define

$$
\begin{aligned}
\hat{a}_t(\boldsymbol{x}^*, \boldsymbol{w}^*) &= \sum_{s \in S_t} \mathbb{P}(y^* \in R_s) \mathbb{1}[l_t^{(F)}(\boldsymbol{x}; 0 | \boldsymbol{x}^*, \boldsymbol{w}^*, c_s) > \alpha], \\
S_t &= \{s \in [|\Omega| + 1] \mid \mathbb{P}(y^* \in R_s) \geq \zeta / (|\Omega| + 1)\}.
\end{aligned}
$$

Then, $\hat{a}_t(\boldsymbol{x}^*, \boldsymbol{w}^*)$ satisfies the following inequality:

$$
|\mathbb{E}_{y^*}[\mathbb{1}[l_t^{(F)}(\boldsymbol{x}; 0 | \boldsymbol{x}^*, \boldsymbol{w}^*, y^*) > \alpha]] - \hat{a}_t(\boldsymbol{x}^*, \boldsymbol{w}^*)| \leq \zeta.
$$

Lemma 3.3 implies that the number of optimization calculations for (3.3) can be further reduced if the error ζ is allowed. In addition, we must emphasize that $\mathbb{P}(y^* \in R_s)$ is often very small for most s when we actually calculate (3.3). Therefore, from these properties, if we apply Lemma 3.3 using a sufficiently small ζ, we can reduce the computational cost of (3.3) significantly with almost no error. Detailed numerical comparisons are provided in Section 5.

4. Theoretical Analysis

In this section, we provide three theorems regarding the accuracy and convergence properties of our methods. First, we define the misclassification loss $e_\alpha(\boldsymbol{x})$ for each $\boldsymbol{x} \in \mathcal{X}$ as follows:

$$
e_\alpha(\boldsymbol{x}) = \begin{cases} \max\{0, F(\boldsymbol{x}) - \alpha\} & \text{if } \boldsymbol{x} \in \hat{L} \\ \max\{0, \alpha - F(\boldsymbol{x})\} & \text{if } \boldsymbol{x} \in \hat{H} \end{cases}.
$$

Furthermore, for theoretical reasons, we assume that the black-box function f follows GP

$\mathcal{GP}(0, k((\boldsymbol{x}, \boldsymbol{w}), (\boldsymbol{x}', \boldsymbol{w}')))$. In addition, for technical reasons, we assume that the prior variance $k((\boldsymbol{x}, \boldsymbol{w}), (\boldsymbol{x}, \boldsymbol{w})) \equiv \sigma_0^2(\boldsymbol{x}, \boldsymbol{w})$ satisfies

$$0 < \sigma_{0,min}^2 \equiv \min_{(\boldsymbol{x},\boldsymbol{w}) \in \mathcal{X} \times \Omega} \sigma_0^2(\boldsymbol{x}, \boldsymbol{w}) \leq \max_{(\boldsymbol{x},\boldsymbol{w}) \in \mathcal{X} \times \Omega} \sigma_0^2(\boldsymbol{x}, \boldsymbol{w}) \leq 1.$$

Note that even if prior variances of the kernel are larger than 1, we can assume that $\max_{(\boldsymbol{x},\boldsymbol{w}) \in \mathcal{X} \times \Omega} \sigma_0^2(\boldsymbol{x}, \boldsymbol{w}) \leq 1$ without loss of generality by standardizing y with an appropriate positive number. Also note that $\sigma_{0,min}^2 > 0$ implies that prior variances are greater than zero, i.e., there is uncertainty at unobserved points. Moreover, let κ_T be the maximum information gain at step T. Note that κ_T is a measure often used to show theoretical guarantee for GP-based active learning methods (see, e.g., (Srinivas et al., 2010)), and can be expressed using mutual information $I(\boldsymbol{y}; f)$ between the observed vector $\boldsymbol{y}$ and f as $\kappa_T = \max_{A \subset \mathcal{X} \times \Omega} I(\boldsymbol{y}_A; f)$. Then, the following theorem regarding accuracy holds:

Theorem 4.1. Let $h \in \mathbb{R}$, $\alpha \in (0, 1)$, $t \geq 1$, and $\delta \in (0, 1)$, and define $\beta_t = 2 \log(|\mathcal{X} \times \Omega| \pi^2 t^2 / (3\delta))$. Moreover, for a user-specified accuracy parameter $\xi > 0$, we define $\eta > 0$ as

$$\eta = \min \left\{ \frac{\xi \sigma_{0,min}}{2}, \frac{\xi^2 \delta \sigma_{0,min}}{8|\mathcal{X} \times \Omega|} \right\}.$$

Then, when Algorithm 1 terminates, with a probability of at least $1 - \delta$, the misclassification loss is bounded by ξ, that is, the following inequality holds:

$$\mathbb{P}\left(\max_{\boldsymbol{x} \in \mathcal{X}} e_\alpha(\boldsymbol{x}) \leq \xi \right) \geq 1 - \delta.$$

Theorem 4.1 does not state whether Algorithm 1 terminates. The following theorem guarantees the convergence property in Algorithm 1:

Theorem 4.2. Under the same setting as described in Theorem 4.1, let $\gamma > 0$ and $C_1 = 2/\log(1 + \sigma^{-2})$. In addition, let T be the smallest positive integer satisfying the following four inequalities:

$$(1)\ \frac{\sigma^{-2} \beta_T^{1/2} C_1 \kappa_T}{T} < \frac{\eta}{2}, \quad (2)\ \frac{\sigma^{-2} C_1 \kappa_T}{T} < \frac{\eta^2}{4},$$

$$(3)\ \frac{C_1 \beta_T \kappa_T}{T} < \frac{\eta^2}{4},$$

$$(4)\ \frac{1}{2} \log \beta_T - \frac{T \eta^2 \sigma^2}{8 C_1 \kappa_T} < \log(|\mathcal{X}|^{-1} 2^{-|\Omega|} \eta \gamma (2\pi)^{1/2} / 2).$$

Then, Algorithm 1 terminates (i.e., $U_T = \emptyset$) after at most T trials when we use the acquisition function $a_t^{(1)}(\boldsymbol{x}^*, \boldsymbol{w}^*)$.

Furthermore, the similar theorem holds if the acquisition function $a_t^{(2)}((\boldsymbol{x}^*, \boldsymbol{w}^*))$ is used. In this study, owing to the practical performance, we modified the original RMILE to

$$\text{RMILE}_t(\boldsymbol{x}^*, \boldsymbol{w}^*) = \max\{\text{MILE}_t(\boldsymbol{x}^*, \boldsymbol{w}^*), \tilde{\gamma} \sigma_t(\boldsymbol{x}^*, \boldsymbol{w}^*)\},$$

where $\text{MILE}_t(\boldsymbol{x}^*, \boldsymbol{w}^*)$ is given by

$$\text{MILE}_t(\boldsymbol{x}^*, \boldsymbol{w}^*) = \sum_{(\boldsymbol{x},\boldsymbol{w}) \in U_t \times \Omega} \mathbb{E}_{y^*}[\mathbb{1}[l_t(\boldsymbol{x}, \boldsymbol{w} | \boldsymbol{x}^*, \boldsymbol{w}^*, y^*) > h]] - |\{(\boldsymbol{x}, \boldsymbol{w}) \in U_t \times \Omega \mid l_t(\boldsymbol{x}, \boldsymbol{w}) > h - \eta\}|.$$

Then, the following theorem holds:

Theorem 4.3. Under the same setting described in Theorem 4.1, let $\gamma > 0$, $\tilde{\gamma} > 0$, and $C_1 = 2/\log(1 + \sigma^{-2})$. In addition, let T be the smallest positive integer satisfying the following five inequalities:

$$(1)\ \frac{\sigma^{-2} \beta_T^{1/2} C_1 \kappa_T}{T} < \frac{\eta}{2}, \quad (2)\ \frac{\sigma^{-2} C_1 \kappa_T}{T} < \frac{\eta^2}{4},$$

$$(3)\ \frac{C_1 \beta_T \kappa_T}{T} < \frac{\eta^2}{4},$$

$$(4)\ \frac{1}{2} \log \beta_T - \frac{T \eta^2 \sigma^2}{8 C_1 \kappa_T} < \log(|\mathcal{X}|^{-1} 2^{-|\Omega|} \eta \gamma \tilde{\gamma} (2\pi)^{1/2} / 2),$$

$$(5)\ \frac{1}{2} \log \beta_T - \frac{T \eta^2 \sigma^2}{8 C_1 \kappa_T} < \log(|\mathcal{X} \times \Omega|^{-1} \eta \tilde{\gamma} (2\pi)^{1/2} / 2).$$

Then, Algorithm 1 terminates (i.e., $U_T = \emptyset$) after at most T trials when we use the acquisition function $a_t^{(2)}(\boldsymbol{x}^*, \boldsymbol{w}^*)$.

The four (resp. five) inequalities in Theorem 4.2 (resp. 4.3) are respectively sufficient conditions for acquisition functions to be bounded as a function of $\sigma_{t-1}^2(\boldsymbol{x}_t, \boldsymbol{w}_t)$ and to be sufficiently small. These assumptions depend on the choice of the kernel function, but most of the practically used kernels, including Gaussian and linear kernels, satisfy the assumptions. In addition, the order of the maximum information gain κ_T is known to be sublinear under mild conditions (Srinivas et al., 2010). Hence, because the order of β_T is $O(\log T)$, there exist positive integers satisfying the inequalities in Theorems 4.2 and 4.3.

5. Numerical Experiments

We confirmed the performance of the proposed method using benchmark functions and infection simulations. Because of space limitation, we provide a part of experimental results in the main text. All experimental results and detail parameter settings are given in the Appendix. The input space $\mathcal{X} \times \Omega$ was defined as a set of grid points that uniformly cut the region $[L_1, U_1] \times [L_2, U_2]$ into 50×50. In all experiments, we used the following Gaussian kernel as the kernel function:

$$k((x, w), (x', w')) = \sigma_f^2 \exp(-\{(x - x')^2 + (w - w')^2\}/L).$$

Table 1. Computation time (second) for the Booth function setting

	Naive	L1	L2	L3 (10^{-4})	L3 (10^{-8})	L3 (10^{-12})
Proposed1_0.01	138505.60 ± 13334.87	7621.59 ± 1166.23	2370.02 ± 586.94	71.16 ± 25.33	80.55 ± 31.37	86.73 ± 35.34
Proposed2_0.01	106306.10 ± 12331.01	5835.06 ± 1028.99	2608.30 ± 976.06	63.14 ± 10.29	72.53 ± 13.99	78.74 ± 16.29

Moreover, we used $L1$-norm as the distance functions between distributions. Furthermore, we considered the following two distributions as the reference distribution $p^*(w)$:

Uniform: $p^*(w) = 1/50$.

Normal:

$$p^*(w) = \frac{a(w)}{\sum_{w \in \Omega} a(w)}, \quad a(w) = \frac{1}{\sqrt{20\pi}} \exp(-w^2/20).$$

Then, we compared the following acquisition functions:

Random: Select (x_{t+1}, w_{t+1}) by using random sampling.

US: Perform uncertainty sampling, i.e., $(x_{t+1}, w_{t+1}) = \text{argmax}_{(x,w) \in \mathcal{X} \times \Omega} \sigma_t^2(x, w)$.

Straddle_f: Perform straddle strategy (Bryan et al., 2006), i.e., $(x_{t+1}, w_{t+1}) = \text{argmax}_{(x,w) \in \mathcal{X} \times \Omega} v_t(x, w)$, where $v_t(x, w) = \min\{u_t(x, w) - h, h - l_t(x, w)\}$.

Straddle_US: Select x_{t+1} and w_{t+1} by using the straddle of $F(x)$ and $\sigma_t(x_{t+1}, w)$, respectively, i.e., $x_{t+1} = \text{argmax}_{x \in \mathcal{X}} v_t^F(x)$ and $w_{t+1} = \text{argmax}_{w \in \Omega} \sigma_t^2(x_{t+1}, w)$, where $v_t^F(x) = \min\{u_t^F(x; \eta) - \alpha, \alpha - l_t^F(x; \eta)\}$.

Straddle_random: Replace the selection method of w_{t+1} in straddle_US with random sampling.

MILE: Perform the original MILE strategy, i.e., (x_{t+1}, w_{t+1}) was selected by using (6) in (Zanette et al., 2018).

Proposed1_0.1: Perform $a_t^{(1)}(\boldsymbol{x}^*, \boldsymbol{w}^*)$ with $\gamma = 0.1$.

Proposed1_0.01: Perform $a_t^{(1)}(\boldsymbol{x}^*, \boldsymbol{w}^*)$ with $\gamma = 0.01$.

Proposed2_0.1: Perform $a_t^{(2)}(\boldsymbol{x}^*, \boldsymbol{w}^*)$ with $\gamma = 0.1$.

Proposed2_0.01: Perform $a_t^{(2)}(\boldsymbol{x}^*, \boldsymbol{w}^*)$ with $\gamma = 0.01$.

Here, for simplicity, we set the accuracy parameter η to zero. Similarly, because of the computational cost of calculating acquisition functions, we replaced $\mathbb{P}(y^* \in R_s) \mathbb{1}[l_t^{(F)}(x; 0 | x^*, w^*, c_s) > \alpha]$ in (3.3) with zero when $\mathbb{P}(y^* \in R_s)$ satisfies $\mathbb{P}(y^* \in R_s) < 0.005$. In other words, we used Lemma 3.3 with $\zeta / (|\Omega| + 1) = 0.005$ to approximate (3.3).

5.1. Synthetic Data Experiments

We confirmed the performance of the proposed method using synthetic functions. We considered the following four functions, which are commonly used benchmark functions (the last one adds -4000 to the original definition):

Booth: $f(x, w) = (x + 2w - 7)^2 + (2x + w - 5)^2$.

Matyas: $f(x, w) = 0.26(x^2 + w^2) - 0.48xw$.

McCormick: $f(x, w) = \sin(x + w) + (x - w)^2 - 1.5x + 2.5w + 1$.

Styblinski-Tang: $f(x, w) = (x^4 - 16x^2 + 5x)/2 + (w^4 - 16w^2 + 5w)/2 - 4000$.

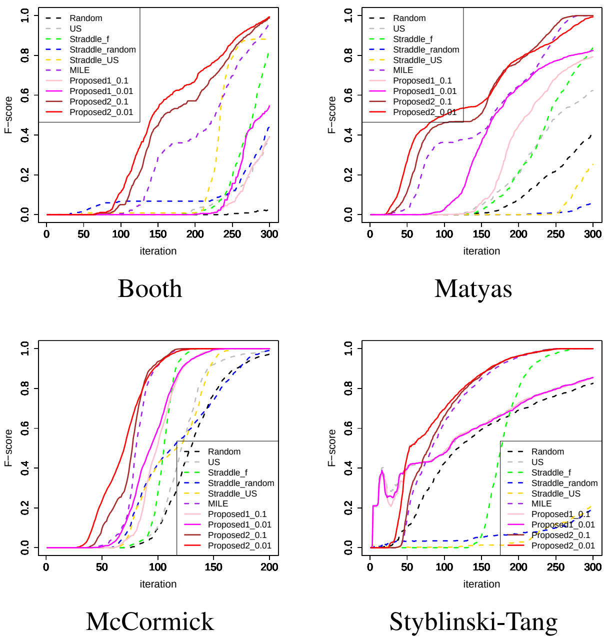
Under this setup, we took one initial point at random and ran the algorithms until the number of iterations reached 300 (or 200), where the parameters used for each experiment are listed in Table 2 in the Appendix. We performed 50 Monte Carlo simulations and obtained the average F-score as follows:

$$\text{F-score} = \frac{2\text{pre} \times \text{rec}}{\text{pre} + \text{rec}}, \ \text{pre} = \frac{|H \cap H_t|}{|H_t|}, \ \text{rec} = \frac{|H \cap H_t|}{|H|}.$$

From Figures 2 and 3, it can be confirmed that our proposed methods outperform other existing methods. On the other hand, in the existing methods, Straddle_f and MILE exhibit high performance, because the MILE acquisition function increases the expected number of (x, w) satisfying $l_t(x, w) > h$. As a result, because $\tilde{l}_t(x, w; \eta)$ and $l_t^{(F)}(x; \eta)$ become large early, the number of elements in H_t also increases early. Similarly, because the Straddle_f acquisition function can efficiently search for (x, w) satisfying $l_t(x, w) > h$ or $u_t(x, w) < h$, the number of elements in H_t also increases efficiently from the same argument as before. Furthermore, when comparing Proposed1 and Proposed2, one of the reasons why the latter exhibits better performance is the fact that RMILE exhibits better performance than uncertainty sampling. Other experiments, a comparison of the difference in γ is described in the Appendix.

5.2. Computation Time Experiments

In this section, we confirmed how much the computation time of (3.2) can be improved by using Lemma 3.1, 3.2 and 3.3. We evaluated the computation time of (3.2) when we



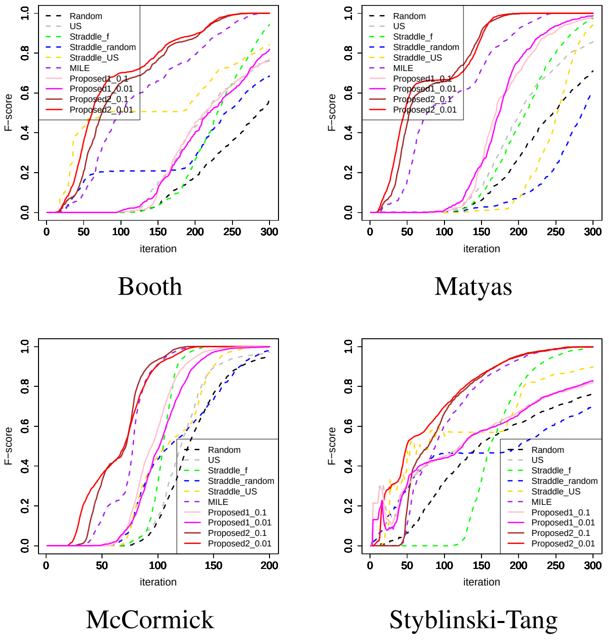
Booth

Matyas

McCormick

Styblinski-Tang

Figure 2. Average F-score over 50 simulations with four benchmark functions when the distance function and reference distribution are $L1$-norm and Uniform, respectively.



Booth

Matyas

McCormick

Styblinski-Tang

Figure 3. Average F-score over 50 simulations with four benchmark functions when the distance function and reference distribution are $L1$-norm and Normal, respectively.

performed the same experiment as in Subsection 5.1 using Proposed1_0.01 and Proposed2_0.01 for the Booth function. The experiments for Matyas, McCormick and Styblinski-Tang functions are described in the Appendix. Here, as for the parameter settings, we considered only the case of $L1$-Normal in Table 2. We compared the computation time of the following six methods for calculating (3.2):

Naive: For each $(\boldsymbol{x}^*, \boldsymbol{w}^*)$, we generate M samples $y_1^*, \dots, y_M^*$ from the posterior distribution of $f(\boldsymbol{x}^*, \boldsymbol{w}^*)$, and approximate (3.2) by

$$\sum_{\boldsymbol{x}\in U_t} \frac{1}{M} \sum_{m=1}^{M} \mathbb{1}[l_t^{(F)}(\boldsymbol{x}; 0|\boldsymbol{x}^*, \boldsymbol{w}^*, y_m^*) > \alpha],$$

where we set $M = 1000$.

L1: Compute (3.2) using Lemma 3.1.

L2: Compute (3.2) using Lemma 3.1 and 3.2.

L3 (10^{-4})**:** Compute (3.2) using Lemma 3.1, 3.2 and 3.3 with $\zeta = (|\Omega| + 1)10^{-4}$.

L3 (10^{-8})**:** Compute (3.2) using Lemma 3.1, 3.2 and 3.3 with $\zeta = (|\Omega| + 1)10^{-8}$.

L3 (10^{-12})**:** Compute (3.2) using Lemma 3.1, 3.2 and 3.3 with $\zeta = (|\Omega| + 1)10^{-12}$.

Under this setup, we took one initial point at random and ran the algorithms until the number of iterations reached to 300. Furthermore, for each trial t, we evaluated the computation time to calculate (3.2) for all candidate points $(\boldsymbol{x}^*, \boldsymbol{w}^*) \in \mathcal{X} \times \Omega$, and calculated the average computation time over 300 trials. From Table 1, it can be confirmed that the computation time is improved as the proposed computational techniques are used. Moreover, comparing L3 (10^{-4}), L3 (10^{-8}) and L3 (10^{-12}), it can be confirmed that the computation time becomes shorter when a large ζ is used. However, it can be seen that the computation time of L3 (10^{-12}) is still very small compared to the computation time of Naive, L1 and L2. Therefore, from $|\Omega| = 50$ and Lemma 3.3, it implies that by using proposed computational techniques, we can improve the computation time significantly even if the error from the true $a_t(\boldsymbol{x}^*, \boldsymbol{w}^*)$ is kept to a very small value such as $51 \times 10^{-12} = 5.1 \times 10^{-11}$.

5.3. Infection Simulations

We compared our proposed method with other existing methods by using the infection control problem (Kermack & McKendrick, 1927). We considered a simulation-based decision-making problem for an epidemic, which aims to determine an acceptable infection rate x under an uncertain recovery rate w with as few simulations as possible. The motivation for this simulation was to evaluate the tradeoff between economic risk and a controllable infection rate. For example, if the infection rate x is minimized by shutting down all economic activities, the economic risk will become extremely high. In contrast, if nothing is done, the infection rate will remain high, resulting in the spread of the disease, and economic risk will still be high. Therefore, we considered finding a target infection rate that can achieve an acceptable economic risk threshold h with a probability of at least α. In this experiment, to simulate epidemic behavior, we used the SIR model (Kermack & McKendrick, 1927). The model computes the evolution of the number of infected people by using an infection rate x and recovery rate w. In our experiment, we considered the infection rate as the design variable x and the recovery rate as the environmental variable w following an unknown distribution. In addition, we regarded economic risk as a black-box function $f(x, w)$. Note that similar numerical experiments were performed in (Iwazaki et al., 2020b) under the setting where the distribution of w, $p^\dagger(w)$, is known. Furthermore, we rescaled the ranges of x and w in the interval $[-1, 1]$. The input space $\mathcal{X} \times \Omega$ is defined as a set of grid points that uniformly cut the region $[-1, 1] \times [-1, 1]$ into 50×50. We used the following economic risk function $f(x, w)$: $f(x, w) = n_{\text{infected}}(x, w) - 150x$, where $n_{\text{infected}}(x, w)$ is the maximum number of infected people in a given period of time, calculated using the SIR model. Note that this risk function was also used by (Iwazaki et al., 2020b), and in this experiment, we used the same function they used in their experiment. Under this setup, we took one initial point at random and ran the algorithms until the number of iterations reached 100. From 50 Monte Carlo simulations, we calculated average F-scores, where we used the following parameters for all problem settings:

$$h = 135,\ \alpha = 0.9,\ \sigma^2 = 0.025,\ \sigma_f^2 = 250^2,\ L = 0.5,$$
$$\beta_t^{1/2} = 4,\ \epsilon = 0.05.$$

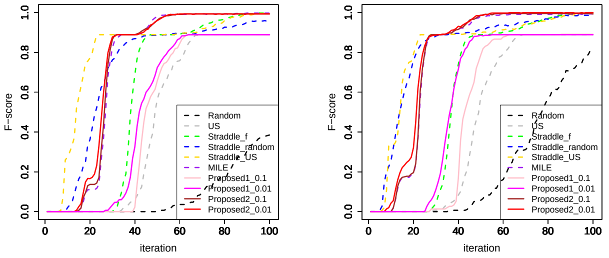
In this experiment, we used the following modified reference function as Normal:

$$p^*(w) = \frac{a(w)}{\sum_{w \in \Omega} a(w)}, \quad a(w) = \frac{1}{\sqrt{0.1\pi}} \exp(-w^2/0.1).$$

From Figure 4, it can be confirmed that Proposed2 and MILE performed better than the others.

6. Conclusion

We proposed active learning methods for identifying the reliable set of distributionally robust probability threshold robustness (DRPTR) measure under uncertain environmental variables. We showed that our proposed methods satisfy theoretical guarantees about convergence and accuracy, and outperform existing methods in numerical experiments.



$L1$-Uniform $L1$-Normal

Figure 4. Average F-score over 50 simulations in the infection control problem with two different settings.

Acknowledgement

This work was partially supported by MEXT KAKENHI (20H00601, 16H06538), JST Moonshot R&D (JPMJMS2033-05), NEDO (JPNP18002, JPNP20006) and RIKEN Center for Advanced Intelligence Project.

References

Beland, J. J. and Nair, P. B. Bayesian optimization under uncertainty. In *NIPS BayesOpt 2017 workshop*, 2017.

Beyer, H.-G. and Sendhoff, B. Robust optimization–a comprehensive survey. *Computer methods in applied mechanics and engineering*, 196(33-34):3190–3218, 2007.

Bogunovic, I., Scarlett, J., Jegelka, S., and Cevher, V. Adversarially robust optimization with Gaussian processes. In *Advances in neural information processing systems*, pp. 5760–5770, 2018.

Bryan, B., Nichol, R. C., Genovese, C. R., Schneider, J., Miller, C. J., and Wasserman, L. Active learning for identifying function threshold boundaries. In *Advances in neural information processing systems*, pp. 163–170, 2006.

Charnes, A. and Cooper, W. W. Chance-constrained programming. *Management science*, 6(1):73–79, 1959.

Fröhlich, L., Klenske, E., Vinogradska, J., Daniel, C., and Zeilinger, M. Noisy-input entropy search for efficient robust bayesian optimization. In Chiappa, S. and Calandra, R. (eds.), *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 2262–2272. PMLR, 26–28 Aug 2020. URL http://proceedings.mlr.press/v108/frohlich20a.html.

Gessner, A., Gonzalez, J., and Mahsereci, M. Active multi-information source Bayesian quadrature. In *Uncertainty in Artificial Intelligence*, pp. 712–721. PMLR, 2020.

Gotovos, A., Casati, N., Hitz, G., and Krause, A. Active learning for level set estimation. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, IJCAI '13, pp. 1344–1350. AAAI Press, 2013. ISBN 978-1-57735-633-2. URL `http://dl.acm.org/citation.cfm?id=2540128.2540322`.

Inatsu, Y., Karasuyama, M., Inoue, K., Kandori, H., and Takeuchi, I. Active learning of Bayesian linear models with high-dimensional binary features by parameter confidence-region estimation. *Neural Computation*, 32 (10):1998–2031, 2020a.

Inatsu, Y., Karasuyama, M., Inoue, K., and Takeuchi, I. Active learning for level set estimation under input uncertainty and its extensions. *Neural Computation*, 32(12): 2486–2531, 2020b.

Inatsu, Y., Sugita, D., Toyoura, K., and Takeuchi, I. Active learning for enumerating local minima based on Gaussian process derivatives. *Neural Computation*, 32(10):2032–2068, 2020c.

Iwazaki, S., Inatsu, Y., and Takeuchi, I. Bayesian experimental design for finding reliable level set under input uncertainty. *IEEE Access*, 8:203982–203993, 2020a.

Iwazaki, S., Inatsu, Y., and Takeuchi, I. Bayesian quadrature optimization for probability threshold robustness measure. *arXiv preprint arXiv:2006.11986*, 2020b.

Iwazaki, S., Inatsu, Y., and Takeuchi, I. Mean-variance analysis in Bayesian optimization under uncertainty. In *International Conference on Artificial Intelligence and Statistics*, pp. 973–981. PMLR, 2021.

Kermack, W. O. and McKendrick, A. G. A contribution to the mathematical theory of epidemics. *Proceedings of the royal society of london. Series A, Containing papers of a mathematical and physical character*, 115(772):700–721, 1927.

Kirschner, J., Bogunovic, I., Jegelka, S., and Krause, A. Distributionally robust Bayesian optimization. In Chiappa, S. and Calandra, R. (eds.), *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 2174–2184. PMLR, 26–28 Aug 2020.

Nguyen, T., Gupta, S., Ha, H., Rana, S., and Venkatesh, S. Distributionally robust Bayesian quadrature optimization. In Chiappa, S. and Calandra, R. (eds.), *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 1921–1931. PMLR, 26–28 Aug 2020.

Oliveira, R., Ott, L., and Ramos, F. Bayesian optimisation under uncertain inputs. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 1177–1184, 2019.

Rahimian, H. and Mehrotra, S. Distributionally robust optimization: A review. *arXiv preprint arXiv:1908.05659*, 2019.

Scarf, H. A min-max solution of an inventory problem. *Studies in the mathematical theory of inventory and production*, 10:201–209, 1958.

Settles, B. Active learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2009.

Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., and De Freitas, N. Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2016.

Srinivas, N., Krause, A., Kakade, S., and Seeger, M. Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, pp. 1015–1022, USA, 2010. Omnipress. ISBN 978-1-60558-907-7. URL `http://dl.acm.org/citation.cfm?id=3104322.3104451`.

Sui, Y., Gotovos, A., Burdick, J., and Krause, A. Safe exploration for optimization with Gaussian processes. In *International Conference on Machine Learning*, pp. 997–1005, 2015.

Sui, Y., Burdick, J., Yue, Y., et al. Stagewise safe Bayesian optimization with Gaussian processes. In *International Conference on Machine Learning*, pp. 4781–4789, 2018.

Toscano-Palmerin, S. and Frazier, P. I. Bayesian optimization with expensive integrands. *arXiv preprint arXiv:1803.08661*, 2018.

Turchetta, M., Berkenkamp, F., and Krause, A. Safe exploration in finite Markov decision processes with Gaussian processes. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pp. 4312–4320, 2016.

Wachi, A., Sui, Y., Yue, Y., and Ono, M. Safe exploration and optimization of constrained MDPs using Gaussian processes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Williams, C. K. and Rasmussen, C. E. Gaussian processes for machine learning. *the MIT Press*, 2(3):4, 2006.

Zanette, A., Zhang, J., and Kochenderfer, M. J. Robust super-level set estimation using Gaussian processes. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 276–291. Springer, 2018.