Information Exploration and Exploitation for Machine Learning with Small Data

by

Shogo Hayashi

A dissertation submitted for the degree of

Doctor of Informatics

Department of Intelligence Science and Technology

Graduate School of Informatics

Kyoto University

Abstract

Recent significant advances in information and communication technologies have facilitated the collection of large amounts of data from Internet of Things (IoT) sensors, online transactions, and other channels. Machine learning technologies are creating a considerable impact in various fields from marketing to science using such big data. In contrast, there are still many scenarios where we can access only a small amount of data, i.e., tens to few hundreds. The small data arises for several reasons. In particular, high data acquisition costs make it difficult to collect a considerable amount of data, for example, during scientific and medical experiments. In other cases, the data related to rare events such as natural disasters requires long collection periods. The number of data newly measured is small, for example, in newly launched websites. The small data limits the effective use of machine learning methods because of a lack of information gained from the data.

In this dissertation, we tackle the learning problems of machines from small data. W supplement the insufficient information with external data and focus on the exploration and exploitation of useful information. Three research topics addressed in this dissertation are classified based on the type of external data. First, we deal with the long-term prediction of time-series data, which tends to be part of the small data scenario because of the long data collection time. Auxiliary data, which is a type of external data, is not directly relevant to tasks but is useful in training models. However, it is not easy to prepare auxiliary data. We propose a framework that explores auxiliary data from the original training data. Next, we extend change-point detection problems with a setting where a learner can access additional data. Additional data, which is another type of external data,

can be used as additional training data. We propose a general framework applicable to different types of data and change-points. The proposed framework determines the next additional data considering the exploration and exploitation of information to identify the solution. Finally, we handle a Bayesian optimization problem. We use additional data in this topic; however, the queries for the additional data are random, in contrast to that for the previous topic and standard Bayesian optimization. We propose two algorithms for scenarios wherein distribution of random queries is known and unknown. Algorithms explore and exploit information when determining the next additional data to be acquired. We demonstrate the effectiveness of the proposed exploration-exploitation approaches with external data via experiments in the small data scenarios using synthetic and real-world data.

Acknowledgements

I would like to express my deep gratitude to everyone I met during my doctor's course. I am grateful to my supervisor, Prof. Hisashi Kashima. His considerable support and scintillating advice were invaluable to my research. I would like to thank the committees for this dissertation, Prof. Akihiro Yamamoto and Prof. Masatoshi Yoshikawa. They reviewed this dissertation in detail and provided insightful comments that contributed to improving this dissertation. My deep appreciation goes to Prof. Yoshinobu Kawahara for providing advice on my career and change-point detection. I would like to thank Mr. Akira Tanimoto for mentoring me during my internship at NEC. I am sincerely thankful to Prof. Motonobu Kanagawa for hosting and advising me when visiting the Max Planck Institute. I would also like to appreciate Prof. Junya Honda for the valuable comments on regret analysis. I would like to thank the members of Kashima–Yamada Laboratory for the stimulating discussions and the time. I appreciate the Japan Society for the Promotion of Science for providing a grant to devote myself to my research. Finally, I am deeply grateful to my family and friends for their moral support and warm encouragement to complete my doctor's course.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

It has become easier to collect a big amount of data in real time through Internet of Things (IoT) sensors, online transactions, and other channels owing to the recent significant advances in information and communication technologies. In addition to the availability of large amounts of data, improvements in computational performance have led to the rapid growth of machine learning, and this is making considerable impacts on a variety of fields ranging from marketing to science. Machine learning is loosely defined as the learning of machines (models) from experiences to predict the future and to gain valuable insights. The experiences are represented as electric records, observations, and data. A wide range of problems have been developed for different types of data and purposes.

- **Classification**: A classification task involves building a classifier that assigns a class label to each data point using a set of input data and class label data. This task is employed in many real-world applications such as image classification, weather prediction, and speech recognition. In weather prediction, which is also known as time-series prediction, the input corresponds to today's weather, humidity, and wind speed, while the labels correspond to tomorrow's weather(i.e., sunny or rainy).

- **Regression**: Regression is a problem of learning relationships between an input variable and a real-valued output variable from a set of input–output observations.

A relationships is used to predict the output from the input. Examples of regression include prediction of stock values and demand forecasting.

- **Clustering**: Clustering methods classify each data point into a specific group. Unlike in a classification task, a set of label data is not provided. The obtained groups are utilized to gain some valuable insights from the data by observing the groups that the data points clustered into. For example, the artificial phylogenies of organisms at the species, genus, or higher level are generated in plant systematics using clustering techniques.

- **Change detection**: While the above problems focus on finding rules that work for the population, change (and anomaly) detection attempts to identify a minority behaving differently from the majority. Changes and anomalies in data often contain valuable information; for example, industrial machine failures, changes in marketing trends, and so on.

We briefly describe the typical problem setting of classification and regression. The entity that performs the task is called *learner*. The learner is given a set of n input–label observations $\mathcal{D} = \{(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}\}_{i=1}^{n}$, called *training data*, where $(x_i, y_i) \overset{\text{i.i.d.}}{\sim} P_{XY}$. $\mathcal{Y} = \{0, 1\}$ for classification and $\mathcal{Y} = \mathbb{R}$ for regression. The goal is to obtain a model $f \in \mathcal{F}$ that predicts y from x well, that is, f should minimize the expected predictive error defined as

$$R(f) = \mathbb{E}_{(X,Y)\sim P_{XY}} \left[l(f(X), Y) \right],$$

where l denotes a loss function as a performance measure quantifying the degree to which f fails to capture the relationship between x and y. However, the learner can access P_{XY} only via training data $\mathcal{D}$. Therefore, the model is obtained, for example, by minimizing

the training loss

$$R_n(f) = \frac{1}{n}\sum_{i=1}^{n} l(f(x_i), y_i).$$

Such a selection of a model f using training data $\mathcal{D}$ is called *training*. The measurement of the model performance using *test data*, which is unseen data, is called *test*.

The spread of machine learning in society has been supported by the ease of access to massive amounts of data. However, despite the rise in such "big data," there are still many scenarios wherein we can access only a small amount of data, i.e., tens to a few hundreds of observations. The "small data" can be attributed to several reasons in many real-world scenarios. Data acquisition is often expensive in terms of financial and temporal costs. Limited budgets allow us to collect only small-sized datasets, and it is specifically the case for survey, medical, and experimental data. In other scenarios, natural disasters such as volcanic eruptions and earthquakes are rare events, which occur a few times annually. However, it is not easy to collect a lot of data over a long period of time, and therefore, the number of data is likely to be small. In the other cases, if the data is just beginning to be observed, its size is small. In newly launched websites, because the time elapsed collecting data is not large, there is an insufficient number of past transactions. This scenario is known as the cold-start problem in the literature of recommendation tasks.

Thus, we are faced with small-data in many real-world situations that results in the challenge of learning from small data. The smallness of data makes the effective use of machine learning methods considerably limited because of the lack of information through the data. The difficulty of the problem is rigorously stated in the statistical learning theory. According to the Vapnik–Chervonenkis (VC) theory [75], with probability at least $1-\delta$, the expected predictive error $R(f)$ of a function $f \in \mathcal{F}$ over a function family $\mathcal{F}$ is upper-bounded by

$$R(f) \leq R_n(f) + O\left(\left(\frac{|\mathcal{F}|_{\mathrm{VC}} - \log\delta}{n}\right)^{\alpha}\right),$$

where $|\mathcal{F}|_{\mathrm{VC}}$ denotes the VC dimension of the function space $\mathcal{F}$ and α represents a sample efficiency constant such that $0.5 \leq \alpha \leq 1$. We can see that a smaller n produces the larger upper bound, and it is difficult to guarantee the good performance of the model.

As stated above, the difficulty in the small data problem lies in the insufficiency of information from small data. To overcome the difficulty, a major approach is to make effective use of external data to supplement the amount of information from small data. External data is classified into two types: auxiliary data and additional data. Auxiliary data is data that can not be directly used as training data but is relevant to the task and useful for improving the performance of machine learning models. Auxiliary data is often available only during training time but not during test time. For example, when the task is to predict the health status of patients via their body conditions such as temperature and blood pressure, medical records maintained by doctors can be used as auxiliary data. Because medical records contain essential information about the health status of patients, it may be useful for training the prediction model. In the literature, there exist several approaches that employ auxiliary data as follows.

- **Semi-supervised learning**: A learner receives labeled training data and unlabeled data. Semi-supervised learning [86] is employed in scenarios where unlabeled data is easily accessible but labels are expensive to obtain. Unlabeled data as auxiliary data can help the learner estimate the population of input data.

- **Transfer learning**: Transfer learning [57] is a framework that transfers knowledge obtained from a source task to the target task. Data from the source task corresponds to auxiliary data. It is assumed that the source task is related to the target task in some sense; there are several variants of transfer learning with respect to the relationship between the tasks.

- **Learning using privileged information**: Learning using privileged information (LUPI) [76, 77] is a learning scheme that utilizes auxiliary data attached with each data point for training a prediction model. The auxiliary data is assumed to be an

intelligent representation of the original data.

To use these auxiliary data approaches effectively, the key or difficulty lies in the preparation of auxiliary data that helps in training models using original data. Unfortunately, we can not expect the use of auxiliary data in general scenarios. In the health prediction example, it may not be easy to collect a set of medical records because of privacy concerns. The other type of external data, additional data is data that a learner can additionally access and use to train the model. Since data acquisition often incurs some costs, the total number of additional data is limited. For example, we may collect more survey data from the survey respondents by incurring costs. Various methods that use additional data have been developed in the machine learning community; these methods are listed below.

- **Bayesian optimization** (BO): BO [66] is a framework for optimizing an unknown expensive-to-evaluate function wherein a learner determines the next input to query the function for its value in an interactive manner. The applications include hyperparameter tuning of machine learning models and chemical compound search.

- **Active learning**: Active learning [64] is a learning scheme where a learner actively selects an unlabeled data point and queries an oracle for its label at a cost to train the model better. The motivation is similar to that of semi-supervised learning, where the labeling cost is high. Active learning is often implemented by crowdsourcing.

The selection of good additional data with a limited number of data acquisitions is important. Some data may be informative for training models, while others may not. Consider the above survey as an example. If the amount of survey data from men is sufficient, women's opinions may provide more information than men's opinions. This would mean that asking for only women's opinions would be more cost-efficient than asking only men or asking both randomly.

In this dissertation, we tackle machine learning with small data based on an external data approach. We provide systematic methodologies that explore and exploit useful information to better train machine learning models in the three research topics. Further,

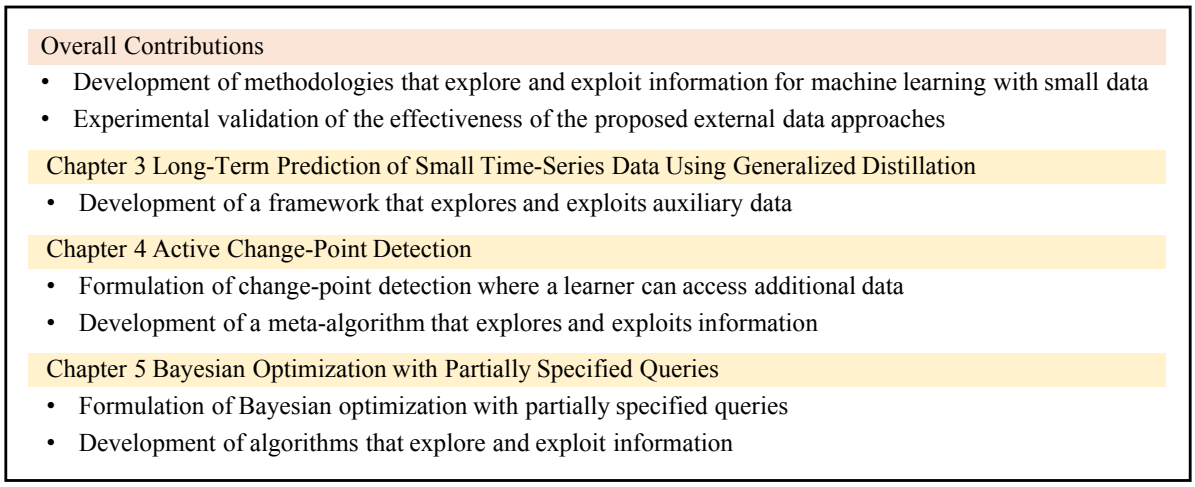


Figure 1.1: Contributions of this dissertation and chapters.

we demonstrate the effectiveness of the methodologies through experimental simulations. The contributions of this dissertation are summarized in Figure 1.1. The relationship among the three topics is illustrated as a taxonomy in Figure 1.2. The remaining chapters of this dissertation are summarized as follows.

Chapter 2 describes general problem settings using external data. We introduce a couple of existing techniques that use auxiliary data and additional data: generalized distillation (GD) and BO.

Chapter 3 discusses the long-term prediction of time-series data in the small data scenario. Long-term prediction is a common and important task to predict time-series data in the long-term from current data. The applications include the task to predict whether a stock price rises in future months given the past-to-current stock prices. Another example is predicting health conditions in a month given the daily heart rate data. The small data scenarios in long-term prediction are likely to occur because the data collection over long periods of time is time consuming and difficult. The long-term prediction of small time-series data is a challenging task, not only because the training sample size is small, but also because the far future is more uncertain than the near future. Auxiliary data possibly improves the performance, but the availability of auxiliary data is not expected in general scenarios because the collection of auxiliary data over long periods of time is

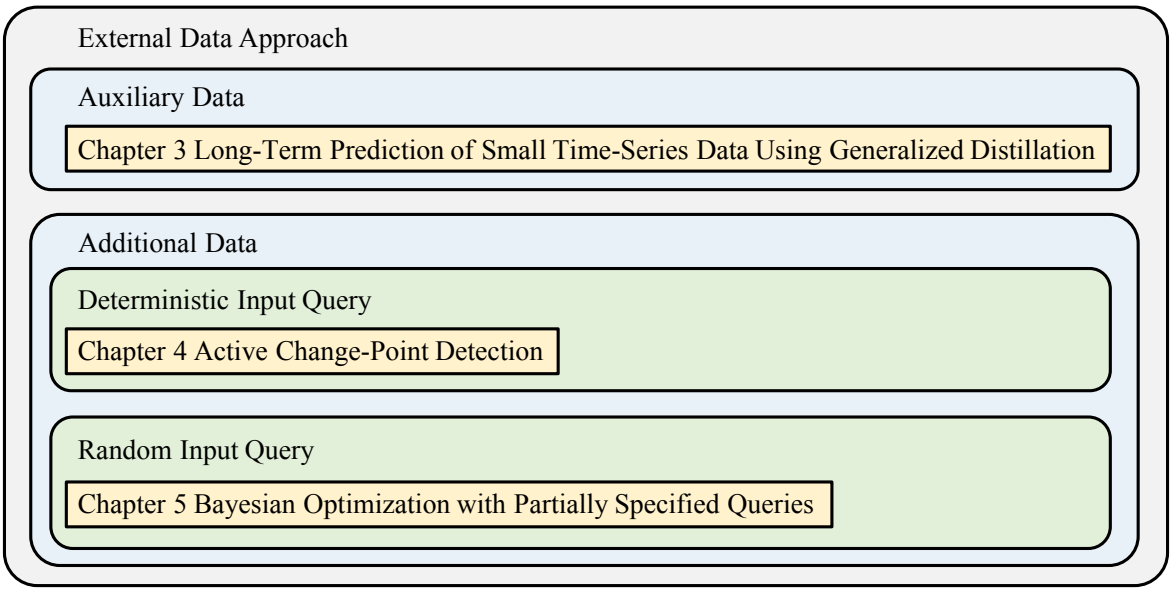


Figure 1.2: Taxonomy of the three research topics ddressed in this dissertation.

not easy. Additional data might be accessible in some scenarios; however, it requires long periods of time to collect additional data because of the problem setting. Therefore, we propose a framework that explores useful auxiliary data from the original time-series data. Hence, the framework does not require a user to prepare any auxiliary data or additional data. The idea is to use middle-time data between the current time and the prediction time as auxiliary data. The middle-time data is available only during the training phase and not during the test phase. Then, the middle-time data is exploited to improve model performance based on GD. We demonstrate that the exploitation of the middle-time data as auxiliary data is effective via experiments using synthetic data and real-world data in a small data scenario.

Chapter 4 extends change-point detection with an additional data setting. Typical change-point detection settings have thus far considered only a static setting, where data is provided with a fixed size. However, when data is not sufficiently provided, change-point detection suffers from poor estimation. We consider change-point detection in an additional data setting, which we call Active Change-Point Detection (ACPD). ACPD is motivated well with important applications in society. For example, in experiments

to find a phase transition, experimenters sequentially measure physical properties of a material at a particular temperature. Because each measurement incurs a cost and the budget for experiments is limited, they would like to examine the phase transition with as few experiments as possible. Therefore, what needs to be addressed is balancing the trade-off between acquiring data at points where there is insufficient data to explore the change-point and acquiring data at points where there is sufficient data to estimate the change-point better. The former and latter are called exploration and exploitation, respectively. We propose a framework that sequentially finds useful additional data to detect change-points while considering the balance between exploration and exploitation. Our idea is to consider the problem as a black-box optimization problem of an unknown change score function. Under this idea, BO is performed with change scores calculated using a particular change-point detection algorithm. This approach can be applied for different types of data and change-points, by appropriately selecting a kernel function and a change-point detection algorithm based on the user's interest. We show that the proposed framework finds change-points with a smaller number of additional data compared to baseline methods using synthetic data and real-world data such as phase transition data and seafloor depth data.

Chapter 5 proposes a novel BO framework that utilizes random queries to reduce the cost of acquiring additional data. In Chapter 4 and standard BO, queries for additional data can be fully specified, i.e., it is deterministic. However, the full specification of queries is costly in some scenarios, for example, when the production of queries is outsourced. Therefore, we instead consider the use of queries which is partially specified for its economical cheapness. We formalize Bayesian Optimization with Partially Specified Queries (BOPSQ), which utilizes partially specified queries to find the maximum input of a black-box function. In BOPSQ, it is assumed that the unspecified part of queries follows a known or unknown distribution, and hence, the full queries are random, which is in contrast to Chapter 4 and standard BO. The trade-off between exploration and exploitation needs to be addressed in BOPSQ. A learner needs to focus on not only exploration to

avoid falling into a local solution but also on exploitation to find the better solution. For the cases of known and unknown distributions, we propose two algorithms that properly handle the trade-off. We further guarantee the performance of the algorithms with regret upper bounds. The experimental results using synthetic data and real-world data indicate the effectiveness of the algorithms.

Chapter 6 presents the concluding remarks of this dissertation. Further, we discuss the limitations and possibilities of machine learning with small data.

Chapter 2

Preliminaries

In this chapter, we present preliminaries for the following chapters. First, for Chapter 3, we describe GD, which is a framework that utilizes auxiliary data to train machine learning models better. Then, for Chapters 4 and 5, we introduce BO, which is a procedure where additional data is sequentially acquired to maximize an unknown function.

2.1 Generalized Distillation

2.1.1 Learning Using Privileged Information

A set of input–label pairs $\{(x_i, y_i) \in \mathcal{X} \times \{0, 1\}\}^n$ is provided in a standard binary classification problem. A learning paradigm that utilizes auxiliary data, called LUPI [77, 76], was proposed. The auxiliary data is specially called privileged information (PI), because it is only available during the training phase as a privilege. PI $x^* \in \mathcal{X}^*$ is associated to each input–label pair (x_i, y_i) and assumed to be some intelligent representation of x_i.

A specific example of LUPI is the prediction of patients' health status based on their body conditions such as temperature and blood pressure. Medical records maintained by doctors correspond to PI. Because the medical records are what doctors, who are experts in medicine, have written about patients' health status from the body conditions, they

can be considered as intelligent representations of the body conditions. Medical records are available during the training phase but not during the test phase. Another example of LUPI is an image classification of numbers wherein input and output data are the noisy image of a number and the number, respectively. One may use noiseless images as PI.

In the problem setting of LUPI, a set of triplets including input, PI, and output $\{(x_i, x_i^\star, y_i) \in \mathcal{X} \times \mathcal{X}^\star \times \{0,1\}\}_{i=1}^n$ is provided. Then, the task is to output a prediction model $f : \mathcal{X} \to \{0,1\}$.

2.1.2 Framework

Distillation [28] is a model compression technique used for the faster inference of deep neural networks. Lopez-Paz et al. [48] incorporated the LUPI setting into distillation and proposed GD. First, in GD, a teacher model $f^* : \mathcal{X}^* \to \mathbb{R}, f^* \in \mathcal{F}^*$, is trained. Then, a student model $f_s : \mathcal{X} \to \mathbb{R}, f_s \in \mathcal{F}_s$, is trained using the teacher model as the final prediction model. The training in GD (for classification tasks) proceeds as follows.

1. Train a teacher model $f^\star \in \mathcal{F}^\star$ using a set of input and PI pairs $\{(x_i^\star, y_i)\}_{i=1}^n$ by solving
$$f^\star = \arg\min_{f \in \mathcal{F}^\star} \frac{1}{n} \sum_{i=1}^n l(y_i, \sigma(f(x_i))) + \Omega(f), \tag{2.1}$$
where $l(y, y') = -y \log(y') - (1-y) \log(1-y')$ represents a cross-entropy loss function, $\Omega(\cdot)$ denotes a regularizer, and $\sigma(x) = 1/(1+\exp(-x))$ represents a sigmoid function.

2. Generate soft targets $\{s_i \mid s_i = \sigma\left(f^\star(x_i^\star)/T\right)\}_{i=1}^n$ using $f^\star$, where $T > 0$ is a temperature hyperparameter.

3. Using an imitation hyperparameter $\lambda \in [0,1]$, pairs of input and output $\{(x_i, y_i)\}_{i=1}^n$, and pairs of input and soft target $\{(x_i, s_i)\}_{i=1}^n$, train a student model $f_s \in \mathcal{F}_s$ by solving
$$f_s = \arg\min_{f \in \mathcal{F}_s} \frac{1}{n} \sum_{i=1}^n \left[(1-\lambda) l(y_i, \sigma(f(x_i))) + \lambda l(s_i, \sigma(f(x_i)))\right]. \tag{2.2}$$

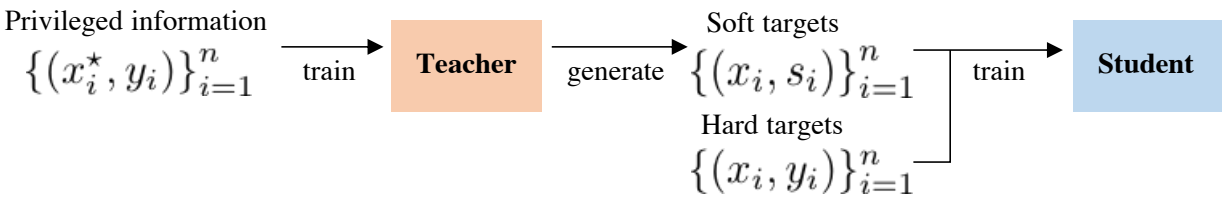


Figure 2.1: Learning scheme of GD using PI.

Figure 2.1 summarizes the learning scheme of GD.

2.1.3 Conditions Under Which Generalized Distillation Works

Lopez-Paz et al. [48] discussed the conditions under which GD works based on the VC theory [75]. The possible conditions are (a) the hypothesis space of the teacher model $|\mathcal{F}_t|_{\mathrm{C}}$ is sufficiently smaller than that of the student model; (b) the approximation error of the teacher model is sufficiently smaller than that of the student model; and (c) the sample efficiency of the student model using GD is greater than the sample efficiency of the baseline model without GD.

2.2 Bayesian Optimization

This section describes the problem setting of BO, Gaussian processes (GPs), which are commonly used as a function prior in BO, and the popular BO techniques.

2.2.1 Black-Box Optimization

Black-box optimization, also known as zeroth-order optimization, is a framework for finding the maximizer of an unknown, i.e., black-box function in an interactive manner. Let $\mathcal{X} \subset \mathbb{R}^d$ be a d-dimensional compact set and $f : \mathcal{X} \to \mathbb{R}$ be an unknown real-valued function. In each iteration, a learner determines an input variable $x \in \mathcal{X}$. The learner then evaluates the function at x and obtains an output $y = f(x) + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma^2)$. The

goal is to find the maximizer $x^* = \arg\max_{x \in \mathcal{X}} f(x)$ with as few evaluations as possible.

BO is an approach for black-box optimization when it is expensive to evaluate an unknown function. The total number of function evaluations is assumed to be less than a few hundreds. A black-box optimization with a GP prior for f is called Gaussian process bandit.

The key notion in the BO problems (or rather bandit problems) is the trade-off between *exploration* and *exploitation* of knowledge. When a learner knows $f(x)$ takes a high value, she may query the function for the value around x by *exploiting* the knowledge. When she does not know what value $f(x)$ takes, she may query the function for the value around x to *explore* the knowledge. Since the number of the function evaluations is limited, there exists a trade-off between them when selecting the next data.

2.2.2 Gaussian Processes

Definition

A GP is a stochastic process that has been widely applied for nonlinear regression and classification. We first define a GP as shown below.

Definition 1 (Gaussian Process [60]) *We call a function $f : \mathcal{X} \to \mathbb{R}$ follows a GP, represented as $f \sim \mathcal{GP}(\mu, k)$, if an output vector $\boldsymbol{f} = (f(x_1), f(x_2), ..., f(x_n))^\top$ of any finite input set $\boldsymbol{x} = (x_1^\top, x_2^\top, ..., x_n^\top)^\top \in \mathcal{X}^n$, $n \in \mathbb{N}$ follows a multivariate Gaussian distribution described as*

$$\boldsymbol{f} \sim \mathcal{N}(\boldsymbol{\mu}, K),$$

where $\mu : \mathcal{X} \to \mathbb{R}$ and $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ denote mean function and covariance function, respectively, and $(\boldsymbol{\mu})_i = \mu(x_i), (K)_{i,j} = k(x_i, x_j)$ for $i, j \in [n]$.

As indicated in Definition 1, the covariance function k characterizes the smoothness of the function or similarity between data points: if covariance $k(x, x')$ between $x, x' \in \mathcal{X}$

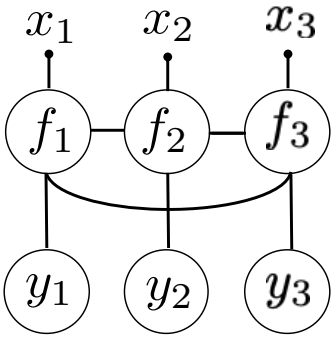


Figure 2.2: Graphical model of a GP for three inputs.

is high, their outputs $f(x)$ and $f(x')$ are highly correlated and vice versa. The graphical model of a GP is illustrated in Figure 2.2.

Covariance Functions

Covariance functions have an important role in characterizing GPs and they are also called kernel functions in the literature of kernel methods. A set of kernel functions have been proposed. The linear kernel defines the covariance by an inner product between two points as

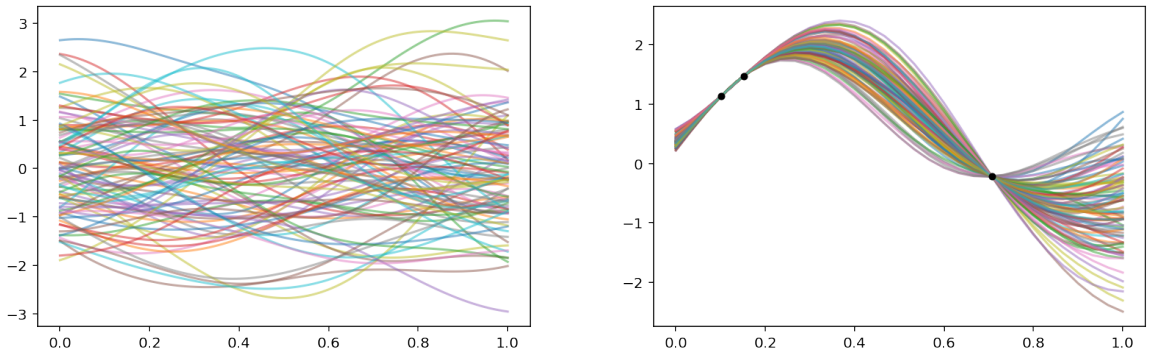
$$k(x, y) = x^\top y + \sigma^2, \tag{2.3}$$

where $\sigma \geq 0$. Here, the function is simply represented by a linear function. The Gaussian kernel, also known as the square exponential kernel, is defined as

$$k(x, y) = \exp\left(\frac{\|x - y\|^2}{2l^2}\right), \tag{2.4}$$

where $l > 0$ is called bandwidth. The Matérn kernel is a class of kernel functions written as

$$k(x, y) = \frac{2^{1-\nu} r^\nu B_\nu(r)}{\Gamma(\nu)}, \tag{2.5}$$



(a) 100 prior sample paths (b) 100 posterior sample paths.

Figure 2.3: 100 sample paths from (a) a GP prior and (b) its posterior.

where $r = \sqrt{2\nu}\,\|\boldsymbol{x} - \boldsymbol{x}'\|/l$, B_ν denotes a modified Bessel function, and $l, \nu > 0$. The hyperparameter ν determines the smoothness of the function, and as $\nu \to \infty$, the Matérn kernel becomes equivalent to the Gaussian kernel.

Posterior

Assume $f \sim \mathcal{GP}(\mu, k)$. Suppose that a set of input–output observations $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ is given. The posterior also follows a GP $f \sim \mathcal{GP}(\mu_n, k_n)$. Here, we have a posterior mean $\mu_n(x)$, posterior variance $\sigma_n^2(x)$, and posterior covariance $k_t(x, y)$ as

$$\mu_n(x) = \mu(x) + \boldsymbol{k}_x^\top \left(K + \sigma^2 I\right)^{-1} (\boldsymbol{y} - \mu(x)), \tag{2.6}$$

$$\sigma_n^2(x) = k_n(x, x), \tag{2.7}$$

$$k_n(x, y) = k(x, y) - \boldsymbol{k}_x^\top \left(K + \sigma^2 I\right)^{-1} \boldsymbol{k}_y, \tag{2.8}$$

where $(\boldsymbol{k}_x)_i = k(x_i, x)$, $(K)_{(i,j)} = k(x_i, x_j)$, I denotes an identity matrix, and $(\boldsymbol{y})_i = y_i$ for $i, j \in [n]$. Figure 2.3 shows 100 sample paths from a GP prior with a zero mean and from the posterior GP with three observations; it indicates that the uncertainty of sample paths decreases given the observations.

Hyperparameter Learning

A set of hyperparameters $\theta \in \Theta$ of a kernel function k_θ have an effect on GP prediction. It is optimized by the type II maximum likelihood estimation, which maximizes the log marginal likelihood as

$$\theta^* = \max_{\theta \in \Theta} \left\{ \log p(\boldsymbol{y}|\boldsymbol{x}, \theta) = -\frac{1}{2}\boldsymbol{y}^\top \left(K + \sigma^2 I\right)^{-1} \boldsymbol{y} - \frac{1}{2}\log\left|K + \sigma^2 I\right| - \frac{n}{2}\log 2\pi \right\}.$$

This optimization can be solved by gradient-based solvers.

As another mean, we consider a fully Bayesian approach; place a prior on θ, and marginalize θ as

$$p(f \mid x, \mathcal{D}) = \int_{\theta \in \Theta} p(f \mid x, \theta, \mathcal{D}) p(\theta \mid \mathcal{D}) d\theta.$$

However, because the integral is often analytically intractable, approximation or Markov chain Monte Carlo methods are used to compute the integral.

2.2.3 Acquisition Functions

BO [66] is a methodology for black-box optimization with an expensive-to-evaluate function. The BO methods typically assume a GP prior on the black-box function $f \sim \mathcal{GP}(\mu, k)$. Without loss of generality, we assume a zero mean $\mu = \mathbf{0}$.

BO methods often use an acquisition function $a : \mathcal{X} \to \mathbb{R}$ that quantifies how much an input point should be evaluated. Suppose a set of observations $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{t-1}$ is given in iteration t. The next input query is determined as

$$x_t = \arg\max_{x \in \mathcal{X}} a(x). \tag{2.9}$$

After observing an output y_t for the query x_t, the data and GP model are updated. The BO methods repeat this procedure until the query budget is exhausted. Algorithm 1

Algorithm 1 Bayesian Optimization

Input: A GP prior $\mathcal{GP}(\mathbf{0}, k)$, a query budget T
Initialize $\mathcal{D}_0 \leftarrow \emptyset$
for $t = 1, 2, \ldots, T$ **do**
  Determine the next input variables $x_t \in \mathcal{X}$ using Equation (2.9)
  Evaluate the function at x_t and observe an output y_t
  Update $\mathcal{D}_{t+1} \leftarrow \mathcal{D}_t \cup \{(x_t, y_t)\}$ and $\mathcal{GP}$
end for

summarizes the framework of the BO methods.

A popular choice for the acquisition function is the Gaussian process upper confidence bound (GP-UCB) [70], which uses a deterministic acquisition function

$$a(x) = \mu_{t-1}(x) + \beta_t^{1/2} \sigma_{t-1}(x), \tag{2.10}$$

where β_t denotes a monotonically increasing deterministic function in t and controls the balance between exploration and exploitation. Expected improvement (EI) [51] is the expected values above the current greatest observed output y_{t-1}^{best}, given as

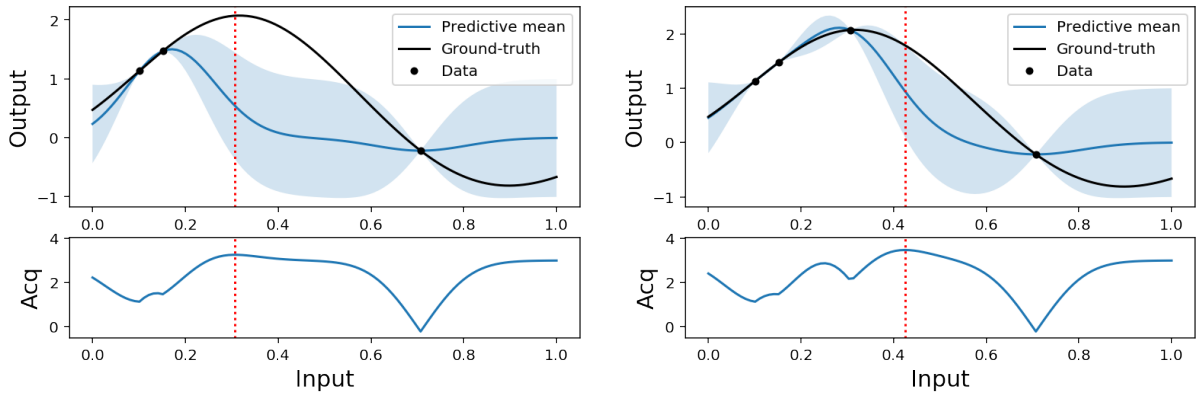
$$\begin{aligned} a(x) &= \sigma_{t-1}(x) \left(\gamma_{t-1}(x) \Phi \left(\gamma_{t-1}(x) \right) + \mathcal{N}(\Phi \left(\gamma_{t-1}(x) \right) \mid 0, 1) \right), \\ \gamma_{t-1}(x) &= \frac{\mu_{t-1}(x) - y_{t-1}^{\text{best}}}{\sigma_{t-1}(x)}, \end{aligned} \tag{2.11}$$

where $\Phi(\cdot)$ denotes the cumulative distribution function of the standard normal. The other popular choice is Thompson sampling [62], which utilizes a posterior sample path as a probabilistic acquisition function written as

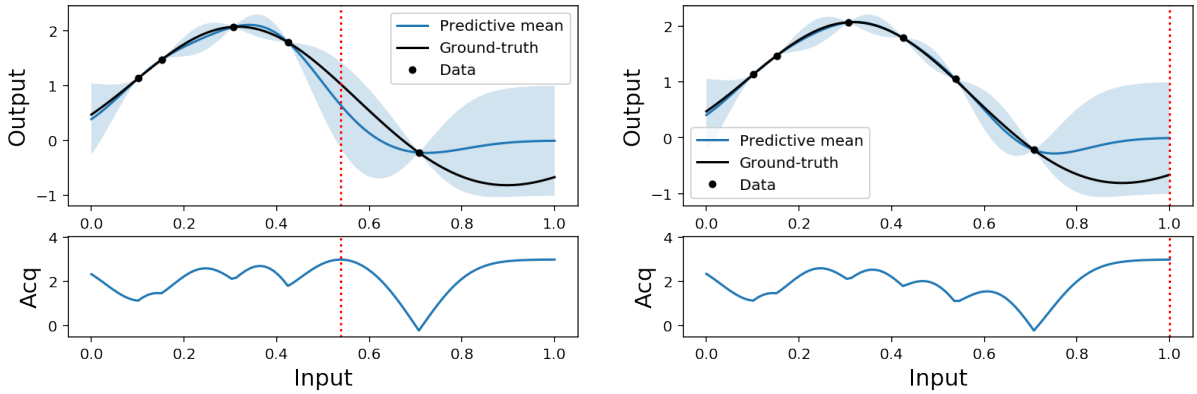
$$a(x) = g_t(x), \tag{2.12}$$

where $g_t \sim \mathcal{GP}(\mu_{t-1}, k_{t-1})$.

Figure 2.4 illustrates the optimization procedure of GP-UCB with $\beta_t = 3^2$ for $t = 1, 2, 3, 4$ over an interval $[0, 1]$. For $t = 1, 2, 3$, it explores around $x = 0.3$ according to



(a) $t = 1$ (b) $t = 2$



(c) $t = 3$ (c) $t = 4$

Figure 2.4: Optimization procedure of GP-UCB with $\beta_t = 3^2$.

the high posterior mean and posterior variance. The new three observations decreased the posterior variance at around $x = 0.3$. Then, $x = 1$ was selected for high posterior variance.

Chapter 3

Long-Term Prediction of Small Time-Series Data Using Generalized Distillation

In this chapter, we tackle the long-term prediction of time-series data, which is likely to suffer from the small-data scenario. While auxiliary data can possibly assist in training machine learning models, its preparation is not necessarily easy. We propose a framework that explores auxiliary data from training data in long-term time-series prediction. The key idea of the proposed framework is to exploit middle-time data between input and output times as auxiliary data, which is available only in the training phase and not in the test phase.

3.1 Introduction

Long-term prediction of time-series is a considerably common and important task. One example is a task to predict whether a stock price will rise in future months given the past-to-current stock price. Another example is a task to predict health status in a month given the daily heart rate data. In typical time-series prediction given time-series data

$\{x_t\}_{t=1}^{L}$, the task is to build a one-step-ahead predictor that predicts x_{t+1} as output from x_t as input. In long-term prediction, the task is to build a long-term-ahead predictor that predicts x_{t+k} from x_t for $k > 1$. Time-series data can be small in two terms of the short length of the time-series and the small number of time series episodes. Long-term prediction tends to result in the small data scenario due to the time and cost of collecting time-series data over time. In addition, because the time-series data points are correlated with each other and not independent and identically distributed (i.i.d.), this is a case where there is substantially smaller data size than in the case of i.i.d. data [8]. Further, the size of data in long-term prediction is smaller than that of the one-step-ahead prediction by $k-1$ because $L-k$ input–output pairs $\{(x_t, x_{t+k})\}_{t=1}^{L-k}$ are available for long-term prediction while $L-1$ input–output pairs $\{(x_t, x_{t+1})\}_{t=1}^{L-1}$ are available for one-step-ahead prediction. In addition to the small data scenario in long-term prediction, the nature of time-series data, where the far future is more uncertain than the near future, makes the long-term prediction of the small time-series data a considerably challenging task.

The difficulty of the problem is rigorously stated in the statistical learning theory, as stated in Chapter 1. The VC theory [75] states that, with probability at least $1-\delta$, the expected predictive error $R(f)$ of a function $f \in \mathcal{F}$ over a function family $\mathcal{F}$ is upper-bounded by

$$R(f) \leq R_n(f) + O\left(\left(\frac{|\mathcal{F}|_{\mathrm{VC}} - \log\delta}{n}\right)^{\alpha}\right), \tag{3.1}$$

where $|\mathcal{F}|_{\mathrm{VC}}$ denotes the VC dimension of the function class $\mathcal{F}$, $R_n(f)$ is the training loss for f with sample size n, and α is a sample efficiency constant such that $0.5 \leq \alpha \leq 1$. Thus, roughly speaking, the second term $O(n^{-\alpha})$ of the upper bound in Equation (3.1) excluding the constant term is the gap between the expected predictive error $R(f)$ and the training error $R_n(f)$, which is the quantity we want to minimize. A smaller second term indicates more effective training. However, this term is 100^{α} times larger when the size

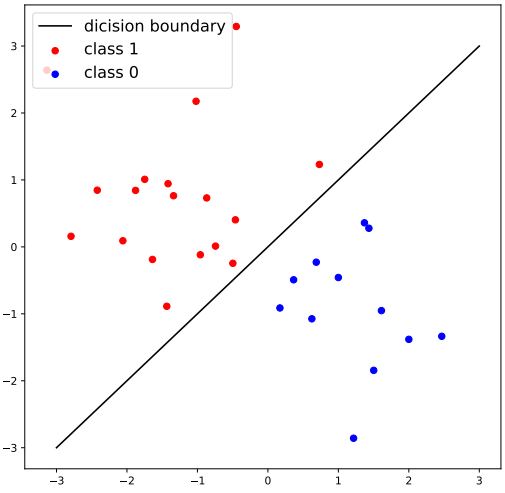


Figure 3.1: Separable case.

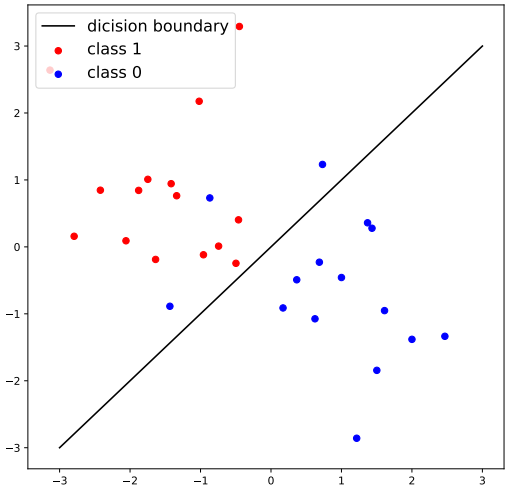


Figure 3.2: Nonseparable case.

of data is $n = 100$ than when $n = 10,000$. This observation indicates that learning from small data is unreliable. The exponential part α represents the convergence rate (sample efficiency constant) as n increases, and in a simple classification problem where the two classes are fully separable such that $a = 1$ as illustrated in Figure 3.1, the convergence rate of the upper bound in Equation (3.1) is $O(n^{-1})$. If the two classes can not be completely separated such that $\alpha = 1/2$ as illustrated in Figure 3.2, the convergence rate is $O(n^{-\frac{1}{2}})$. Therefore, to obtain the same upper bound for both besides the constant term, they need sample sizes of $n = 100$ and $n = 10,000$, respectively; thus, we see that the difference in the sample size varies exponentially. Figures 3.1 and 3.2 show that the simple decision boundary separating the training sample classes coincides with the ground-truth decision boundary as n increases when the two classes are separable: the learning in the separable case is easy. However, it is not always the case when the two classes are nonseparable; i.e., the learning in the nonseparable case is not easy. The above consideration suggests that the upper bound of the expected predictive error for small data can be decreased if the sample efficiency constant α successfully approaches 1 or if the nonseparable learning problem is transformed into a separable learning problem.

Recent studies for such small-sample-size problems focus on exploiting auxiliary data [32].

A promising framework using such auxiliary data to improve the sample efficiency α to 1 is LUPI [77, 76] and GD [48]. The LUPI framework has demonstrated its effectiveness in the small data scenario by exploiting PI to assist the training of machine learning models. However, we may not assume the availability of PI in all situations. For example, in Chapter 2.1, we introduced the prediction of patients' health status as a LUPI problem, where medical records can be used as PI; however, medical records are not always available because they contain private information. The preparation of PI itself is as difficult as the preparation of data.

In this chapter, we propose a novel framework for the long-term prediction of small-sized time-series data based on GD. In particular, the framework solves the difficulty in preparing PI using the property of long-term prediction that middle-time data between input time t and prediction time $t + k$ is unavailable during the test phase but available during the training phase. This property is inherent only in long-term prediction. Our main idea is using middle-time data as PI. The middle-time data is somehow related to input and prediction time data and can be more powerful than the original input in prediction for time-series data with the Markov property. Based on the GD framework, we exploit middle-time data to improve the predictive performance of machine learning models in the small data scenario. We demonstrate the effectiveness of the proposed framework on both synthetic data and real-world data in the small-data scenarios. Experimental results indicate that the proposed framework performs well, particularly when the task is difficult and has high input dimensions.

The rest of this chapter is organized as follows. Section 3.2 introduce the problem setting of long-term time-series prediction. In Section 3.3, we propose a new approach for training long-term prediction models by exploiting middle-time data. Section 3.4 summarizes related works and discusses relations between them and the proposed framework. In Section 3.5, we present the experimental results on synthetic data and real-world data to demonstrate the effectiveness of our proposed framework. Finally, in Section 3.6, we provide the concluding remarks in this chapter with some possible future research direc-

tions.

3.2 Problem Setting

We define the long-term time-series prediction problem as a task given an input time-series sequence of L length $\{x_t \in \mathcal{X}\}_{t=1}^{L}$ and an output time-series sequence $\{y_t \in \mathcal{Y}\}_{t=1}^{L}$ to build a prediction model that predicts y_{t+k} from $\{x_\tau\}_{\tau=1}^{t}$ at time t for $k > 1$. The input domain $\mathcal{X}$ can be multi-dimensional, continuous, and discrete. The output domain $\mathcal{Y}$ is one-dimensional and can be continuous or discrete. In the patients' health status prediction example, the input includes two-dimensional real values of body temperature and blood pressure, PI is multi-dimensional discrete values of a medical record written in the natural language, and the output is a binary value of the health status (healthy or unhealthy).

In the long-term time-series prediction problem, it is assumed that $k > 1$. However, a very large k may make the problem too hard because x_t and y_{t+k} may be independent. Hence, we assume that k is not too large, and therefore, y_{t+k} can be predicted from x_t.

It is important to note that This problem setting is different from the multi-step time-series prediction problem [14, 78], where the goal is to build a model that predicts an output sequence $\{y_\tau\}_{\tau=t+1}^{t+k}$ from $\{x_\tau\}_{\tau=1}^{t}$ at time t, although long-term prediction is referenced to as multi-step prediction in some literatures. The difference between the two problem settings is illustrated in Figure 3.3.

3.3 Proposed Framework

The key idea of the proposed framework is to exploring further information from middle-time data $\{x_\tau\}_{\tau=t+1}^{t+k-1}$ between input time t and prediction time $t+k$. Then, middle-time data is utilized as PI in the GD framework. In fact, middle-time data satisfies the condition of PI: unavailability during the test phase. This idea is based on the inherent

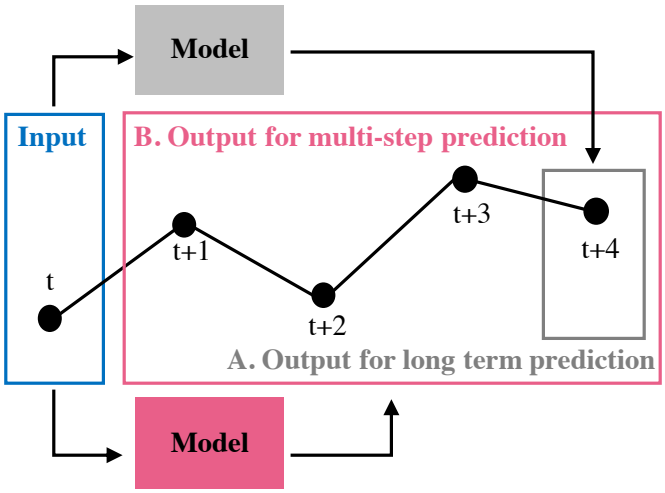


Figure 3.3: Difference between the problem settings of the long-term prediction (marked as "A") and multi-step prediction (marked as "B").

nature of long-term prediction and does not appear in the one-step-ahead prediction setting. Further, middle-time data has some relationship between input data and output data. In the case of time-series data with the Markov property, middle-time data can be considered a smart representation of input data because middle-time data has stronger correlation with the output data than the input data; i.e., middle-time data is more useful than the input data for the prediction. Hence, middle-time data can be used as PI. In the GD framework, it is assumed that the data satisfy the i.i.d condition. Time-series data does not satisfy the i.i.d condition and data points have correlations with each other. Nevertheless, the discussion on the GD framework with time-series data would be possible as well with i.i.d. data under some assumptions such as the stationarity of time series data, the existence of spectral density, and its finite maximum eigenvalue [8]. Here, the actual sample size of time-series data reduces because of its autocorrelation.

Figure 3.4 illustrates the idea of the proposed framework. In particular, when L-length training time-series data $\{x_t\}_{t=1}^{L}$ is provided and we use middle-time data from time $t+1$ to $t+k-1$ as PI, we have a set of triplets of input, PI, and output $\{(x_t, x_t^\star, x_{t+k})\}_{t=1}^{L-k}$. Next, a teacher model is trained using PI and output data $\{(x_t^\star, x_{t+k})\}_{t=1}^{L-k}$. The teacher

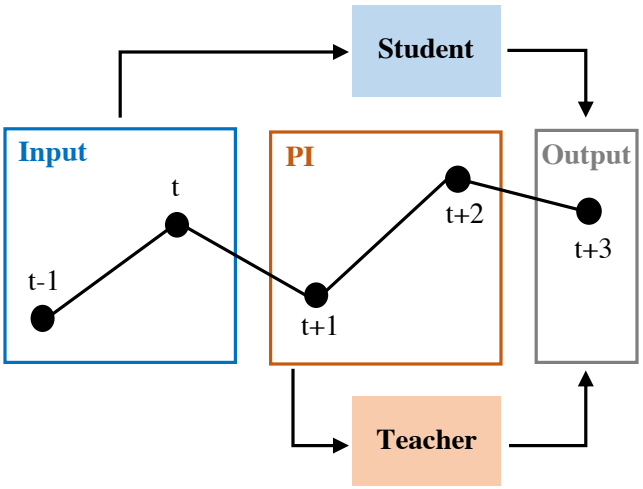


Figure 3.4: Idea of the proposed framework for long-term prediction.

model computes soft targets from input data $\{s_t \mid s_t = \sigma\left(f^\star(x_t^\star)/T\right)\}_{t=1}^{L-k}$, where T is a temperature hyperparameter. Finally, a student model is trained using a set of input–output pairs $\{(x_t, x_{t+k})\}_{t=1}^{L-k}$ and a set of PI–output pairs $\{(x_t, s_{t+k})\}_{t=1}^{L-k}$. Algorithm 2 demonstrates the pseudo-code for binary classification. For regression problems, the loss function is replaced with an appropriate one such as the mean square error.

When prediction time k and the number of dimensions of data d are large, the use of all middle-time data as PI may worsen the prediction performance of the teacher model because of the curse of dimension of $(k-2)d$. A bad teacher produces a bad student in the GD framework. Therefore, we consider using a part of the middle-time data as PI. We simply consider (A) one-step-ahead data x_{t+1}, (B) data at the middle time $x_{\lceil (t+k)/2 \rceil}$, and (C) one-step-behind data from the prediction time x_{t+k-1} for the time-series data with the Markov property, as illustrated in Figure 3.5, according to the conditions under which GD works in Section 2.1.3. (C) may decrease the approximation error of the teacher model because of the strong correlation between (C) and output data; however, the sample efficiency constant may not improve because of the weak correlation between (C) and input data because the problem for the teacher model to predict the output value from (C) is different from the original problem for the student model. (A) may not decrease

Algorithm 2 Proposed Framework for Long-term Prediction of Small Time-Series Data

Input:

Input sequence: $\{x_t\}_{t=1}^{L}$
Output sequence: $\{y_t\}_{t=1}^{L}$
Rule to generate a PI sequence: R
Prediction time: k
Temperature hyperparameter: $T > 0$
Imitation hyperparameter: $\lambda \in [0, 1]$

Output: Student model $f_s \in \mathcal{F}_s$

1: Generate a PI sequence $\{x_t^\star\}_{t=1}^{L}$ using rule R from input sequence $\{x_t\}_{t=1}^{L}$.
2: Generate a set of input–output pairs $\{(x_t, y_{t+k})\}_{t=1}^{L-k}$ and a set of PI–output pairs $\{(x_t^\star, y_{t+k}\}_{t=1}^{L-k}$.
3: Train a teacher model $f^\star \in \mathcal{F}^\star$ using $\{(x_t^\star, x_{t+k})\}_{t=1}^{L-k}$ as

$$f^\star = \arg\min_{f \in \mathcal{F}^\star} \frac{1}{L-k} \sum_{t=1}^{L-k} l(x_t^\star, \sigma(f(x_t^\star))) + \Omega(\|f\|).$$

4: Generate soft targets $\{s_t \mid s_t = \sigma\left(f^\star(x_t^\star)/T\right)\}_{t=1}^{L-k}$ using $f^\star$.
5: Train a student model $f_s \in \mathcal{F}_s$ with a set of input–output pairs $\{(x_t, y_t)\}_{t=1}^{L-k}$ and a set of input–soft-target pairs $\{(x_t, s_t)\}_{t=1}^{L-k}$ as

$$f_s = \arg\min_{f \in \mathcal{F}_s} \frac{1}{L-k} \sum_{t=1}^{L-k} [(1-\lambda) l(x_{t+k}, \sigma(f(x_t))) + \lambda l(s_{t+k}, \sigma(f(x_t)))].$$

6: **return** Student model $f_s \in \mathcal{F}_s$.

the approximation error of the teacher model because of the weak correlation between (A) and output data; however, the sample efficiency constant may improve owing to the strong correlation between (A) and input data because the problem to predict the output value from (A) is similar to the original problem. The property of (B) may lie between those of (A) and (C). In summary, there may be a trade-off between the prediction accuracy of the teacher model and the similarity between the original problem for the student model and the problem for the teacher model. In Section 3.5.1, we empirically examine differences in prediction performance among the middle times.

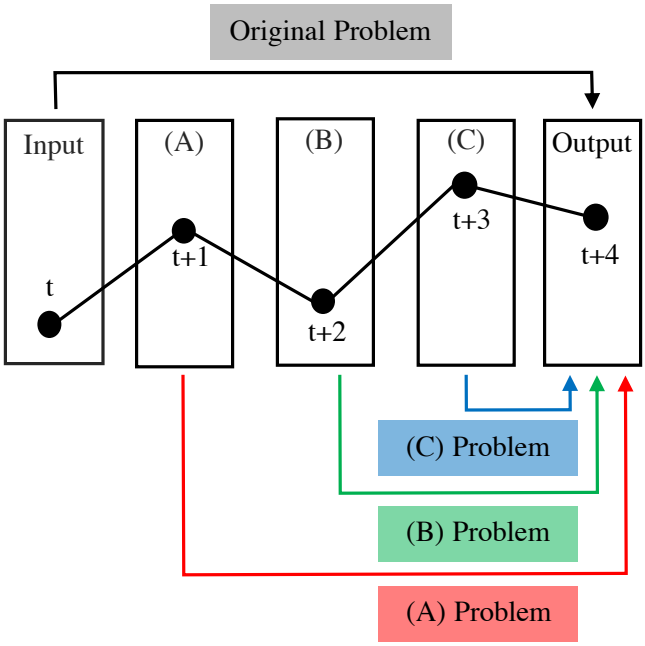


Figure 3.5: Difference of the middle-time data: (A) one-step-ahead data x_{t+1}, (B) data at the middle time $x_{\lceil (t+k)/2 \rceil}$, and (C) one-step-behind data from the prediction time x_{t+k-1}.

3.4 Related Work

In this section, we review existing works from several viewpoints and discuss the relations between them and the proposed framework.

Time-series prediction [9] is an attractive task in machine learning and data mining, and it has been extensively studied. The task is to predict the future of sequence data. The applications range widely in various fields including finance, climatology, medicine, and control. Several approaches have been proposed, including autoregressive models [1, 30] and support vector machine models [10, 52].Recently, neural network models such as recurrent neural networks [13, 42] have gained attention for their promising results.

For multi-step time-series prediction, a method that transforms the original data into some meaningful and interpretable entities following the principle of justifiable granularity based on hidden Markov models was proposed [23]. A multi-integration strategy was proposed [54], wherein a prediction is generated by integrating predictions of multiple

predictive models. In this strategy, k-nearest-neighbor-based least-squares support vector machines were employed for multiple predictors, and an autoregressive model was used for the integration. An optimally pruned extreme learning machine was applied to the multi-step time-series prediction [22]. A combination method of a direct prediction strategy and a sophisticated input selection criteria was proposed [69]. Here, input selection criteria were optimized using a proposed selection strategy that combines forward selection, backward elimination or pruning, and forward–backward selection. The performance of nonlinear autoregressive models that use neural networks with exogenous inputs was examined [35]. These methods are designed for multi-step prediction. Therefore, we do not compare them in the experimental part.

The problem of small data can be improved by utilizing auxiliary data, which is available only in the training phase but not in the test phase. Jonschkowski et al. [32] systematically summarized learning schemes using auxiliary information. One representative approaches is transfer learning [57], where auxiliary data is called source data and it is associated with the target task in some sense. In multi-view learning [71], auxiliary data is associated with input data.

GD was originally proposed by Lopez-Paz et al. [48] for unifying distillation [28] and LUPI [76, 77]. Distillation is a model compression technique that trains a neural network with shallower and narrower layers using hard targets (training labels) and soft targets generated by another neural network with deeper and wider layers. It is assumed that soft targets have valuable information, e.g., the distance from each sample to the decision boundary, which can accelerate the learning rate of the smaller network. Vapnik and Vashist [77], Vapnik and Izmailov [76] proposed a new learning paradigm LUPI. In LUPI, a student model is trained with a set of triplets: input, PI, and output. Note that PI is considered a smart expression of knowledge for each instance provided by an intelligent teacher. Further, GD unifies distillation and LUPI, where a student model is trained with hard targets and soft targets provided by a teacher model trained with PI.

Ao et al. [5] applied GD to transfer learning, where soft targets are generated using

other domain data as PI weighted by imitation hyperparameters, and support vector machines are employed as a student model. That formalization enables the learning of the imitation hyperparameters.

In addition, GD is applied to time-series data. A noise-robust automatic speech recognition system using acoustic speech data was proposed [49]. An acoustic model based on deep neural networks and hidden Markov models is used, wherein the input is speech data and the output is its state. The teacher model is trained with noiseless speech data, whose soft targets are used to train the student model with noisy speech data. Further, GD is extended and applied to speech normalization tasks [33, 34]. Although these studies dealt with time-series data, our work is the first to apply GD to the long-term prediction of time-series data. Our work does not assume that PI is given, and the proposed framework generates it from original training data by utilizing the property of long-term prediction.

3.5 Empirical Evaluation

In this experiment, we focus on the prediction of time-series data in small-data scenarios. We conduct a preliminary experiment using simple synthetic data to examine how different PI works. Then, we evaluate the proposed framework using synthetic datasets and real-world datasets with different dimensions. The experimental results indicates that the proposed framework improves prediction accuracy when the dimension of data is relatively high.

3.5.1 Preliminary Experiment

The use of all middle-time data may cause a poor performance of the teacher model and the proposed framework because of the curse of dimensionality. The number of dimensions of all middle-time data is $d(k-1)$, which is large in the long-term prediction scenario. Further, because of the autocorrelation of time-series data, the use of all middle-time data is redundant. Therefore, in our experiments, we use part of middle-time data and

compare them at different times.

In this experiment, we generate a set of 50 sequences of length 25, $\{\{x_t^i\}_{t=0}^{t=24}\}_{i=1}^{50}$, using a random walk with the Markov property written as

$$x_{t+1} = x_t + \epsilon,$$

where $\epsilon \sim \mathcal{N}(0, 0.3)$ and $x_0 \sim \mathcal{N}(0, 1)$. Each sequence is labeled y_i according to whether the value at $t = 20$ is positive or negative, as shown in Figure 3.6. We set the task to predict y given x_0 and evaluate each method using 1000 test sequences. We conduct each experiment 30 times.

We set a logistic regression model as a teacher model and a student model in the proposed framework as

$$p(y \mid x) = \sigma(wx + w_0).$$

Parameters $w, w_0 \in \mathbb{R}$ are estimated by the maximum likelihood estimation. The prediction is performed as

$$y = I(p(1 \mid x) > 0.5),$$

where $I(\cdot)$ denotes the indicator function. The training data, which is a set of triplets of input, PI, and output $\{(x_0^i, x^{i\star}, y_i)\}_{i=1}^{50}$. We set the time range for the middle-data used as PI x_l^i from $l = 0$ to 24. We set the temperature hyperparameter $T = 1.0$ and the imitation hyperparameter $\lambda = 0.5, 1.0$. Learning with $\lambda = 1.0$ corresponds to learning with only soft targets.

Figure 3.7 shows the prediction error of the proposed method with different middle-time from 0 to 24 for PI. The proposed method with $\lambda = 0.5$ and 1.0 improved the mean prediction error of the baseline model (a simple logistic regression model without the proposed framework). The PI indexed with $l = 3, 4$, which is relatively close to

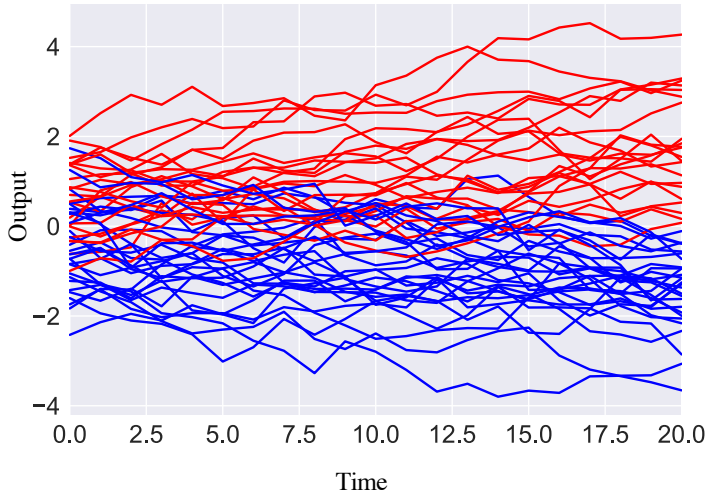


Figure 3.6: Random walk data. Red and blue curves indicate class 1 and 0, respectively.

the input time, achieved the best results. This may be because the middle-time data is relatively close to the input time and has a strong correlation with input data as discussed in Section 3.3. The mean minus one standard deviation of the accuracy of the proposed method when $l \leq 15$ are better than that of the baseline model but not when $l > 15$ because of the large standard deviation. This is probably because the uncertainty of PI and the variance of data increased as l increased.

3.5.2 Data Description

We conduct experiments on both synthetic data and real-world data. We create a Mackey–Glass dataset using a chaotic differential equation, and we use the PM2.5 data observed in Beijing. Using the two datasets, we attempt three experimental settings. The experimental settings are summarized in Table 3.1.

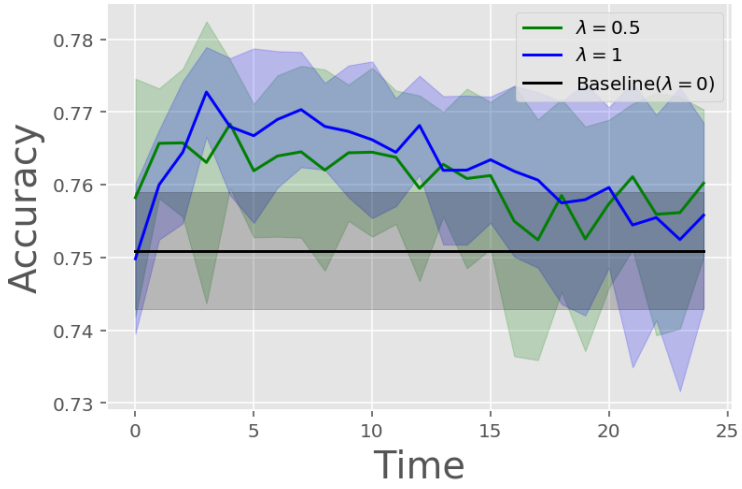


Figure 3.7: Test prediction error when changing middle-time data used as PI in the random walk data. Solid and shaded lines represent the mean and standard deviation of the 30 trials, respectively.

Table 3.1: Experimental setup.

Data	Input length (student)	Input dimensions (student)	Input length (teacher)	Input dimensions (teacher)	Prediction time	Training time-series length	Validation time-series length	Test time-series length
Mackey–Glass	4	4	2	2	8	100	100	10000
PM2.5	(3,7,10,13,16)	(33,77,110, 143,176)	2	22	15	110	100	184

Mackey–Glass Data

We generate one-dimensional chaotic time-series data called Mackey–Glass data using a differential equation described as

$$\frac{dx(t)}{dt} = -ax(t) + \frac{bx(t-t_0)}{1+x(t-t_0)^c}. \tag{3.2}$$

The increment at time t depends on the linear term of the state at t and the nonlinear term of the state at $t-t_0$. In this experiment, we set $a = 0.1, b = 0.2, c = 10, t_0 = 16$, and $x(t_0) = 0.9$. The generated sequence in the discretized time in Figure 3.8 is periodic but it changes slightly. We obtain a binary-labeled sequence as an output sequence based

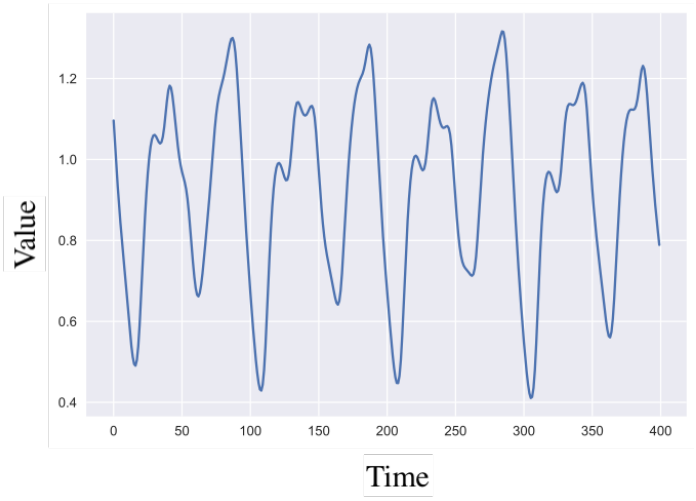


Figure 3.8: Mackey–Glass data generated by Equation (3.2).

on whether the k-step-ahead value is greater, $y_t = I(x_t \leq x_{t+k})$, and we set $k = 8$. The time-series is split into a 100-length training sequence, a 100-length validation sequence, and a 1000-length test sequence such that they do not overlap. We then take a four-length input sequence $(x_{t-3}, x_{t-2}, x_{t-1}, x_t)^\top$ as the input of the student model and a two-length PI sequence as the input of the teacher model.

Beijing PM2.5 Data

As the real-world data to demonstrate the performance of the proposed framework in a real setting, we use the Beijing PM2.5 dataset [1] where the task is to predict the concentration of PM2.5 in the air in the long-term. The long-term prediction of PM2.5 is important because PM2.5 affects the planning of roads and public transportation systems such as trains. This dataset is eight-dimensional time-series data containing the concentration of PM2.5 in the air, humidity, and other measures, observed every hour from 2010 to 2014 in Beijing. We preprocess the data as follows. We transform the one-dimensional wind direction that takes four values $\{East, West, South, North\}$ into a four-dimensional binary vector for

[1] https://archive.ics.uci.edu/ml/datasets/Beijing+PM2.5+Data

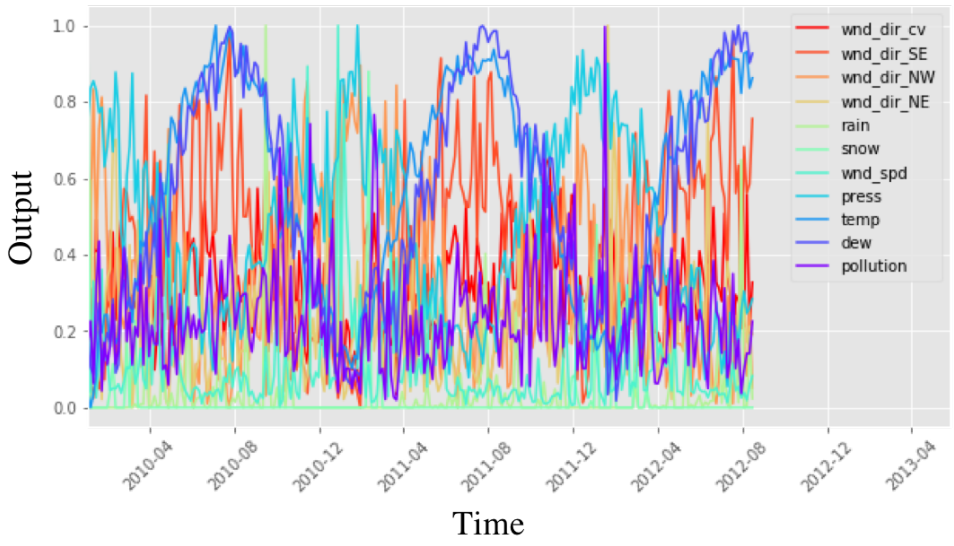


Figure 3.9: PM2.5 data, where poullution indicates the concentration of PM2.5.

each direction. We compute the mean value of every four days to reduce the size of the time-series. We further normalize them into the range $[0, 1]$ using min-max normalization: $(x - x_{\min})/(x_{\max} - x_{\min})$. Figure 3.9 shows the data after preprocessing, where pollution indicates the concentration of PM2.5. The temperature and other parameters vary with a yearly cycle, but the noise in all features is large. In addition to Mackey–Glass data, we obtain a binary-labeled sequence as an output sequence based on whether the k-step-ahead values of the concentration of PM.25 are greater. We set $k = 15$, which corresponds to the prediction in a couple of months. The time-series is split into a 100-length training sequence, a 100-length validation sequence, and a 184-length test sequence so that they do not overlap. The input length of the teacher model is set to 2. To examine the effect of the input dimensions for the student model, we attempt different input lengths of the student model as summarized in Table 3.1.

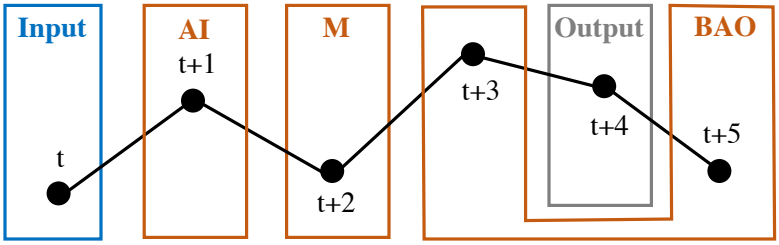


Figure 3.10: Illustration of the types of PI used in the experiments: AI (data after the input data), M (data at the middle-time between the input and the output data), and BAO (data before and after the output data).

3.5.3 Specifications of Model and Privileged Information

We use logistic regression models as the student and teacher models. The models are trained to predict output y_{t+k} using input x_t. We may use one-step-ahead prediction models by iteratively repeating the one-step-ahead prediction k times. However, this approach accumulates the prediction error exponentially because of the iterative prediction [78]. Thus, we employ the model that directly predicts y_{t+k} from x_t. The parameters are determined by maximum likelihood estimation. We add $L1$ norm $\|w\|^1$ or $L2$ norm $\|w\|^2$ as a regularizer. We compare three types of PI: data just after input data (AI), data at middle-time (M) between input and output data, and data before and after output data (BAO). Figure 3.10 illustrates AI, M, and BAO. The hyperparameters, i.e., the weight of the regularizer, the type of PI, the imitation hyperparameter, and the temperature hyperparameter, are determined using validation data. Because the proposed method is designed to improve the prediction accuracy of prediction models, we do not compare other prediction models except for the baseline model in this experiment.

3.5.4 Results and Discussion

Table 3.2 shows prediction accuracies of the methods for the Mackey–Glass data. The proposed method does not improve the baseline model. The baseline model could predict the data well with an accuracy of 0.971 because the Mackey–Glass data is periodic and

Table 3.2: Prediction accuracy of the proposed method and baseline model for the Mackey–Glass data.

Baseline Model	Proposed Method
0.971	0.932

one-dimensional, i.e., easy to predict. The numbers of the input dimensions of the student model (baseline model) and the teacher model are 4 and 2, respectively, and their hypothesis spaces are small. In particular, their VC dimensions are 5 and 3, respectively. Therefore, the teacher model may not have sufficinet room to improve the student model because of the small prediction error of the student model.

Figure 3.11 shows the prediction accuracy of the proposed method and the baseline model when changing the input length $(3, 7, 10, 13, 16)$. The proposed method succeeded in improving the baseline model except when the input length was 3. Since the PM2.5 data is 11-dimensional and the number of the input dimensions is $11 \times$ (input length), the VC dimension of the linear model is $11 \times$ (input length) $+ 1$. When the input length is 3, the VC dimensions of the student model and the teacher model are 34 and 23, respectively. Since the ratio of these values are $1/3$, which is comparable to the Mackey–Glass data, it is likely that the prediction accuracy of the teacher model is insufficient to improve the student model. When the input length is further increased, the difference between the VC dimensions of the teacher model and the student model becomes larger. As a result, the conditions (a), (b), and (c) in Section 2.1 are satisfied, and therefore, the proposed framework worked effectively. In addition, taking the AI, which uses the data just after the input data as PI, tended to produce better results. This is likely because the AI is highly correlated with the input data, similar to the results of the preliminary experiment in Section 3.5.1.

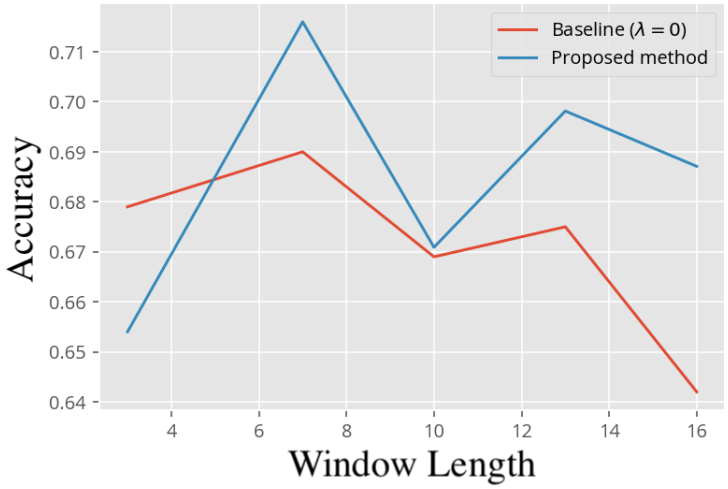


Figure 3.11: Prediction accuracy of the proposed method and the baseline model when changing the input length $(3, 7, 10, 13, 16)$.

3.6 Conclusion

The amount of time-series data for long-term prediction tends to be small because of the high cost of data collection, which causes a poor performance of prediction models. In this chapter, we proposed a novel framework for the long-term prediction of small-sized time-series data. Our main idea is to explore and exploit middle-time data between input time and prediction time, which is available only in the training phase due to the property of long-term prediction. The proposed framework uses the middle-time data as PI in the GD framework to train the prediction model better. The experimental results using the synthetic datasets and the real-world datasets showed that the proposed framework outperformed the baseline method, particularly when the data was multi-dimensional and the hypothesis space was large. The experimental results for comparing different middle-time data suggested that the middle-time data relatively close to the input time worked well and the teacher models with high prediction accuracy were not necessarily good to train the student models.

We conclude this chapter with several possible future works. First, we need further

examinations of more complex and difficult cases with higher-dimensional time-series data such as movie data. According to both the conditions when the GD framework works and the experimental results, the proposed framework will perform better for such data. Considering the recent outstanding successes of deep neural networks, it would be interesting to see how the proposed framework performs for temporal neural networks such as recurrent neural networks.

Another important direction is to develop methods to learn hyperparameters from only training data without validation data because it is not desirable to split a small amount of data into a training dataset and a validation dataset for tuning. A thorough investigation of datasets of various types and sizes would yield more insights about when the proposed framework works well, and this would considerably help us address machine learning tasks with small data.

Chapter 4

Active Change-Point Detection

In the previous chapter, we tackled the small data problem based on the approach that explores and exploits auxiliary data. This chapter handles change-point detection in the small data scenario. We consider solving the small data problem by employing the other approach using additional data. We extend change-point detection in the additional data setting where a learner sequentially determines an input query and observes an output by paying additional cost. We propose a general meta-algorithm that automatically balances exploration and exploitation of information to detect the change-point, which is applicable to various types of data and change-points. The main idea of the proposed meta-algorithm is to regard the problem as a black-box optimization of an unknown change score function.

4.1 Introduction

The problem of detecting abrupt changes in data is called change-point detection [7, 25], and this is closely related to concept drift [19], event detection [24], and time-series segmentation [38]. It has been enthusiastically studied in data mining and industry, and it covers a broad range of data types such as sensor data [31] and dynamic network data [82], among others. Its applications include fault detection [37], network-intrusion detection [84], and trend change detection [47]. Typical change-point detection problems assume

that parts of time-series data are sequentially observed in an online manner or that the entire data is provided at once in a offline manner.

However, the data acquisition cost in change-point detection is often high. The small number of data makes change-point detection difficult. One example is found in material science. Imagine a physical experiment that attempts to detect a phase transition temperature of a material (Figures 4.1, 4.2, and 4.3). A phase transition is a sudden change in a physical property (e.g., density, energy, electric resistance, and specific heat) and a phase (e.g., gel-sol, solid-liquid, and nematic-isotropic) at a particular temperature. Finding phase transitions is important for developing new materials, and they can be viewed as a change-point detection problem. Phase transitions are examined via real experiments and simulations, wherein an experimenter sets a material at a particular temperature and observes its physical property. Such experiments require financial or temporal resources to collect data.

Another example is found in geoscience. Studying the geography of the seafloor, which has rugged landscapes such as ocean trenches (Figure 4.9(a)), is important for understanding ocean currents and other phenomena. Finding such trenches corresponds to change-point detection. Measuring the depth of the sea requires a considerable amount of resources.

Therefore, we consider the use of additional data at additional cost to improve the performance of change-point detection in the small data setting, which has not been considered in existing literature. In this chapter, we propose ACPD, a novel active learning problem of change-point detection. The goal is to detect a change-point in a black-box expensive-to-evaluate target function; however, unlike traditional change-point detection problems, we actively obtain data by querying the target function for its outputs. We aim to determine an effective sequence of input queries to find the change-point using as few queries as possible because of the high data acquisition cost. In the material science example, the input and the output correspond to the temperature and the physical property, respectively. In the geography example, they correspond to the location (longitude

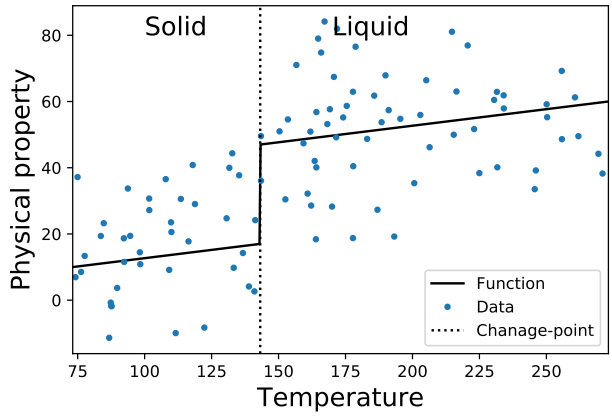


Figure 4.1: First-order phase transition from solid to liquid state at 143.15 (unit) and 100 randomly sampled observations.

and latitude) and the depth, respectively. ACPD does not consider the input is time.

We propose a simple and general solution to ACPD, which relies on neither the underlying data structures nor the definitions of change-points. Our solution is a meta-algorithm based on an idea that we consider ACPD as black-box optimization of a change score function. It reuses an existing change-point detection algorithm to compute change scores from data. Further, it employs a BO technique to determine the next input to find where the change score is high. Our empirical results using synthetic datasets and real-world datasets such as material science data and seafloor depth data that include different types of data and change-points, demonstrate the query efficiency of the proposed framework.

The remainder of this chapter is organized as follows. Section 4.2 briefly reviews the basic settings of offline change-point detection based on which we formalize the ACPD problem. Then, we develop a novel meta-learning framework using the BO method in Section 4.3. In Section 4.4, we empirically demonstrate the query efficiency of the proposed framework in comprehensive settings. Section 4.5 summarizes the related work. Section 4.6 concludes this chapter.

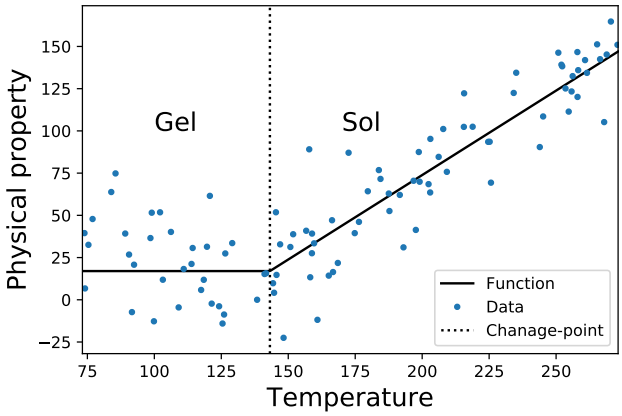


Figure 4.2: Second-order phase transition from gel to sol at 143.15 (unit) and 100 randomly sampled observations.

4.2 Problem Setting

In this section, we describe the problem setting of the standard offline change-point detection and its typical approach. Then, we state the problem setting of ACPD.

4.2.1 Change-Point Detection

There exist a variety of problem settings of change-point detection for the definitions of data and change-points. For simplicity, we review the standard (passive) problem setting of one-dimensional change-point detection with a change-point in an offline setting [7, 21].

Let $f : \mathcal{X} \to \mathbb{R}$ be an unknown function parameterized by a piecewise-constant parameter $\theta \in \Theta$. We assume a domain $\mathcal{X}$ is an interval in $\mathbb{R}$. θ changes from θ_1 to θ_2 at a change-point $x^{\mathrm{cp}} \in \mathcal{X}$: $f(x) = f(x \mid \theta_1)$ if $x < x^{\mathrm{cp}}$, otherwise $f(x) = f(x \mid \theta_2)$. Suppose n observations $\mathcal{D} = (X, Y) = \{(x_i, y_i) \in \mathcal{X} \times \mathbb{R}\}_{i=1}^{n}$ from the function are given, where the index i is aligned in the ascending order of $\mathcal{X}$ (i.e., $x_i \leq x_j$ for $1 \leq i < j \leq n$) and the output contains noise $y_i = f(x_i \mid \theta) + \epsilon_i, \epsilon_i \sim \mathcal{N}(0, \sigma^2)$.

We define a change-point in observations $x^{\mathrm{cp}'} \in X$ as the point at which the function mechanism or the model parameter θ changes. Thus, $y_i = f(x_i \mid \theta_1) + \epsilon_i$ for $x_i < x^{\mathrm{cp}'}$ and $y_i = f(x_i \mid \theta_2) + \epsilon_i$ for $x^{\mathrm{cp}'} \leq x_i$. Typical settings deal with temporal data, i.e., x_i

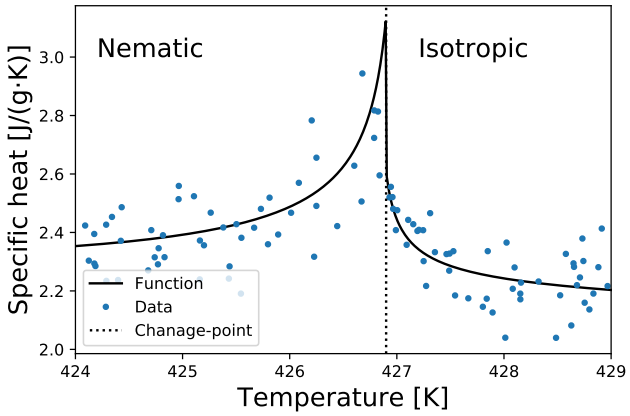


Figure 4.3: Nematic-isotropic phase transition of the CBO11O material at 426.9 [K] [63] and 100 randomly-sampled observations.

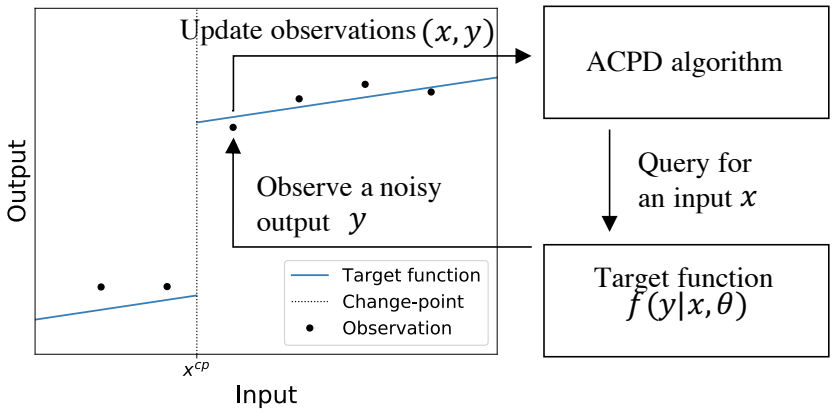


Figure 4.4: Overview of the active change-point detection procedure.

corresponds to a time index that we do not assume in this chapter. We assume that x_i corresponds to a temperature value, y_i corresponds to the measured value of a physical property, and their relationship is governed by some physical law f with some noise (Figures 4.1, 4.2, and 4.3).

In change-point detection problems, the goal is to find the change-point in observations $x^{\text{cp}'} \in X$. A typical approach is to model a *change score function* $s_{\mathcal{D}} \colon X \to \mathbb{R}$, which quantifies how f "changes" over inputs. The change-point is estimated as the point that

maximizes the change score

$$\hat{x}^{\mathrm{cp}'} = \arg\max_{x \in X} s_{\mathcal{D}}(x). \tag{4.1}$$

The change score function is designed such that its scores reflect the type of change-points we want to detect. A possible choice is the maximum likelihood function, which is given as

$$s_{\mathcal{D}}(x) = \max_{\theta_1, \theta_2} \ln \left[\prod_{j | x_j < x} p(y_j \mid x_j, \theta_1) \prod_{k | x \leq x_k} p(y_k \mid x_k, \theta_2) \right], \tag{4.2}$$

where $p(y \mid x, \theta)$ denotes the probability density function of a normal distribution with mean $f(x \mid \theta)$ and variance σ^2. The change score is computed according to how the model with parameters fits the data.

4.2.2 Active Change-Point Detection

For simplicity, we assume one-dimensional continuous input and output variables, and there exists only one change-point as the point with a sudden change in our problem setting as stated before; however, the problem setting can be extended to other types of inputs, outputs, and change-points, e.g., multi-dimensional inputs, outputs and multiple change-point cases, which we will show in the experiments.

In contrast to the passive change-point detection problem, a learner has no control over inputs to the target system f, and the ACPD problem allows the learner to interact with the target system by actively selecting the inputs to be investigated.

The goal of ACPD is to estimate the change-point $x^{\mathrm{cp}} \in \mathcal{X}$ in the function (not in observations) by querying the target function f in an iterative manner. At each iteration, a learner first determines the next input query based on the past observations, provides an input query $x \in \mathcal{X}$ to the target function, and then observes the corresponding output y. We have a limited budget $B \in \mathbb{N}$, which is the maximum number of queries we can

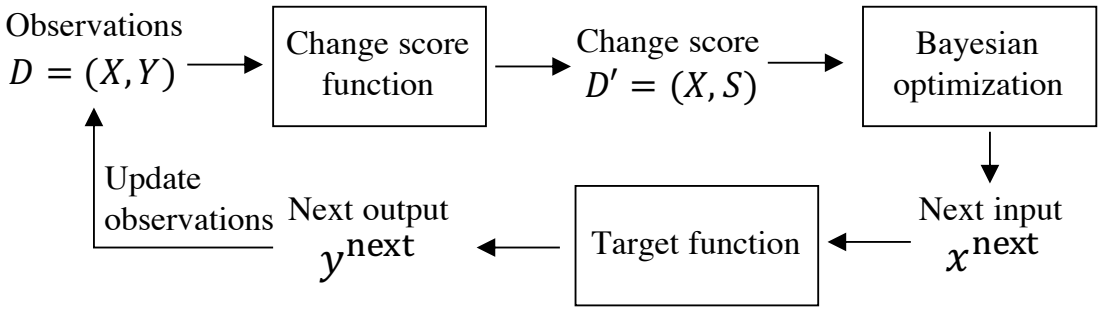


Figure 4.5: Proposed Meta-ACPD framework.

make in total; therefore, an ACPD algorithm is required to suggest an effective sequence of input queries to find the change-point based on past observations. An overview of the ACPD procedure is illustrated in Figure 4.4.

4.3 Proposed Framework

There are many possible definitions of change-points and change score functions depending on target applications. Therefore, instead of designing a solution specialized for a particular choice of data and change-points, we propose Meta-ACPD, a simple and general ACPD framework that is applicable to a wide variety of data and change-points.

Our key idea is to view the ACPD problem as a black-box optimization problem of a change score function. As a summary of the meta-algorithm, Meta-ACPD utilizes an existing change-point detection method specified by a user to compute change scores and then perform BO [51, 67] on the change scores to decide the next input that would maximize the change score. At each iteration, Meta-ACPD uses a change-point detection method to obtain change scores S for data points X observed so far. Meta-ACPD then estimates the change scores over the input space using a GP with the computed change scores, and it determines the next query input x^{next} such that it maximizes the change score using an acquisition function. Meta-ACPD iterates this procedure until the given budget runs out and then outputs the final change-point estimate $\hat{x}^{\text{cp}}$ using the change-

point detection algorithm. Our proposed procedure (for one-change-point functions) is illustrated in Figure 4.5 and Algorithm 3.

In the setting of the offline change-point detection, we see that we typically estimate the change-point as the data point that maximizes some change score in a set of n observations $\mathcal{D} = (X, Y) = \{(x_i, y_i)\}_{i=1}^n$. In ACPD, we attempt to find a change-point in a domain $\mathcal{X}$ such as $\mathbb{R}$. If we had an infinitely large number of observations $\mathcal{D}^\infty$ over the entire domain $\mathcal{X}$, we could estimate the change-point in the same way as the passive offline change-point detection, i.e., by finding the maximizer in the set of computed change scores. Then, the change score function $s_{\mathcal{D}^\infty}$ could be regarded as a function that maps from $\mathcal{X}$ to $\mathbb{R}$. However, we can only access a finite number of observations $\mathcal{D}$. Therefore, we consider estimating $s_{\mathcal{D}^\infty}$ from $\mathcal{D}$, and Equation (4.1) can be regarded as a black-box optimization of the unknown change score function $s : \mathcal{X} \to \mathbb{R}$. However, in addition to the fact that the observations are finite and noisy, the change scores depend on the selection of the observations; further different sets of observations produce different change scores. To take the uncertainty of the computed change scores into consideration, we model the change score function with a probabilistic model, a GP. That model enables us to utilize BO to determine the next input for finding the maximum change score.

Given n observations $\mathcal{D} = (X, Y) = \{(x_i, y_i)\}_{i=1}^n$, a change-point detection algorithm selected by a user is applied to $\mathcal{D}$ to obtain a set of change scores $\mathcal{D}'$ as

$$\mathcal{D}' = \{(x_i, s_i) \mid s_i = s_{\mathcal{D}}(x_i)\}_{i=1}^n = (X, S). \tag{4.3}$$

We assume that a black-box change score function s follows a GP $s \sim \mathcal{GP}(\mathbf{0}, k)$ with a particular kernel function k. Given the computed change scores $\mathcal{D}'$, the posterior mean $\mu_{\mathcal{D}'}$ and variance $\sigma^2_{\mathcal{D}'}$ are obtain using Equations (2.6) and (2.7), respectively. Finally, BO is performed using a particular acquisition function as indicated in Equation (2.9), to determine the next input. By evaluating the target function at the next input, Meta-ACPD observes the corresponding output y^{next}. The above iteration terminates when the query

Algorithm 3 Meta-ACPD

Input:
A change-point detection algorithm and its change score function: s
An acquisition function: a
A query budget: B
An initial set of observations: $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}$
Output: The change-point estimate $\hat{x}^{\mathrm{cp}}$
for $t = 1, 2, \ldots, B$ **do**
  Apply s to $\mathcal{D}$ and obtain change scores $\mathcal{D}'$ (Equation (4.3))
  Determine the next input x^{next} using a (Equation (2.9))
  Observe the output y^{next} by evaluating the target function at x^{next}
  Update the observations $\mathcal{D} \leftarrow \mathcal{D} \cup \{(x^{\mathrm{next}}, y^{\mathrm{next}})\}$
end for
return The change-point estimate $\hat{x}^{\mathrm{cp}}$

budget runs out or some particular condition is satisfied. Consequently, Meta-ACPD estimates the change-point using the change-point detection algorithm. Alternatively, one can estimate the change-point as the point with the maximum posterior mean

$$\hat{x}^{\mathrm{cp}} = \arg\max_{x \in \mathcal{X}} \mu_{\mathcal{D}'}(x). \tag{4.4}$$

One of the significant advantages of the proposed framework is that it depends on neither the target data types nor the definitions of the change-points because of the idea that ACPD is considered as black-box optimization of a change score function. It utilizes an existing change-point detection algorithm to compute one-dimensional change scores, which allows a user to select a proper change-point detection algorithm based on the change-point type of interest. Since it determines the next input using a BO technique, a user can select a proper kernel function based on the type of data. Hence, it would work for different types of inputs, outputs, and change-points, such as multi-dimensional input–output, noise-level changes, and multiple change-points.

Further, the application of BO in the proposed framework can be expected to improve the accuracy of the regression of the target function (not the change score function) and the search for the change-point. When adopting the likelihood as the change score definition,

wherein the regression of the target function is performed, it is necessary to improve the accuracy of the regression. The posterior variance (Equation (2.7)) is typically used inside an acquisition function such as GP-UCB (Equation (2.10)) and EI (Equation (2.11)). This posterior variance is determined only by input data $\{x_i\}_{i=1}^n$. Querying for points with a large posterior variance corresponds to a typical active learning method that queries points with high uncertainty, uncertainty sampling [44]. Thus, performing BO in the proposed framework improves the regression accuracy of the target function by active learning.

Change scores computed from the data can be different when the data is updated. For two sets of observations $\mathcal{D}_n = \{(x_i, y_i)\}_{i=1}^n$ and $\mathcal{D}_{n+1} = \{(x_i, y_i)\}_{i=1}^{n+1}$, it is possibly that $s_{\mathcal{D}_n}(x_i) \neq s_{\mathcal{D}_{n+1}}(x_i)$ for $i = 1, ..., n$. In the standard BO, it is assumed that the input–output data does not varies. It can be considered that when the amount of data is sufficiently large, the effect of updating the data is small; however, especially in the initial stage when the amount of data is small, the change scores are likely to vary. The next input to be evaluated determined using such unreliable change scores is at the risk of being incorrect. However, the proposed framework, which internally performs BO, prioritizes the exploration of the entire input domain over the use of the obtained change scores in the initial stage when the amount of data is small, and this produces data that are relatively uniformly distributed in the input domain. Empirically, the change scores computed using evenly distributed data in the input domain are more similar to the change scores computed using a large amount of data compared to those computed using unevenly distributed data in the input domain. Hence, the effect of updating data is relatively not serious for the proposed framework. In the experimental part in Section 4.4, we report that the proposed framework avoids this problem by performing exploration, although the exploitation approaches tend to fall into local solutions in the initial stages when the data size is small.

4.4 Experiments

For a comprehensive study of the empirical performance of the proposed approach, we study change-point detection accuracy under measurement cost limitations using several target functions with different types of data and change-points.

4.4.1 Target Functions

Functions with one change-point

We first introduce four real-valued functions with a one-dimensional input and a change-point. As simple bench-marking settings, we use

- a noncontinuous piecewise-linear function $\mathrm{PT}_{\mathrm{bias}}$ (Figure 4.1)

$$f(x) = \begin{cases} 0.1x + 30 & (x < 143.15) \\ 0.1x + 60 & (\text{otherwise}). \end{cases}$$

- a continuous piecewise-linear function $\mathrm{PT}_{\mathrm{slope}}$ (Figure 4.2)

$$f(x) = \begin{cases} 0.1x + 30 & (x < 143.15) \\ x + 147 & (\text{otherwise}). \end{cases}$$

These two functions are considered a first-order phase transition and a second-order phase transition, respectively.

Further, we use a five-dimensional-output piecewise-linear function MO defined as

$$f(x) = \begin{cases} ax + b & (x < -130) \\ ax + b + 50 & (\text{otherwise}), \end{cases}$$

where $a = (0.26, -1.29, 0.49, -1.12, -0.45)^\top$ and $b = (1.70, 0.79, 0.33, 0.45, -0.37)^\top$.

As a more realistic scenario, we use the nematic-isotropic phase transition function of the CBO11O material referred to as NI (Figure 4.3), which is a regression result in the real experiment [4, 63]. The function is

$$f(x) = \begin{cases} 2.13 - 2.99\left(\frac{x}{427.03} - 1\right) + 0.0162\left|\frac{x}{427.03} - 1\right|^{-0.51} & (424 \leq x < 426.9) \\ 2.13 - 2.99\left(\frac{x}{426.82} - 1\right) + 0.006\left|\frac{x}{426.82} - 1\right|^{-0.51} & (426.9 \leq x \leq 429) \end{cases}.$$

Gaussian noise is added to the output of each function : $y = f(x) + \epsilon$, $\epsilon \sim \mathcal{N}(0, \sigma^2)$, where the standard deviation is set to $\sigma = 20$ for $\mathrm{PT}_{\mathrm{bias}}$, $\mathrm{PT}_{\mathrm{slope}}$, and MO, and $\sigma = 0.1$ for NI.

Multiple-change-point function

Our proposed framework can be generalized to target functions with multiple change-points. We use a piecewise-constant function with three change-points MCP (Figure 4.6). The function is defined as

$$f(x) = \begin{cases} 9.34 & (x < -143.5) \\ 0.94 & (-143.5 \leq x < -77.7) \\ 9.45 & (-77.7 \leq x < -41.9) \\ 4.30 & (\text{otherwise}). \end{cases}$$

Gaussian noise $\epsilon \sim \mathcal{N}(0, 1^2)$ is added to its output.

Multiple-input function

Another possible extension is multi-dimensional-input functions. In this setting, we focus on finding steep depth changes in a seafloor depth dataset, which is important for sea floor surveys. We use a seafloor depth dataset[1] for the area around Hokkaido, Japan,

[1] Area 0450-09 (https://www.geospatial.jp/ckan/dataset/1976)

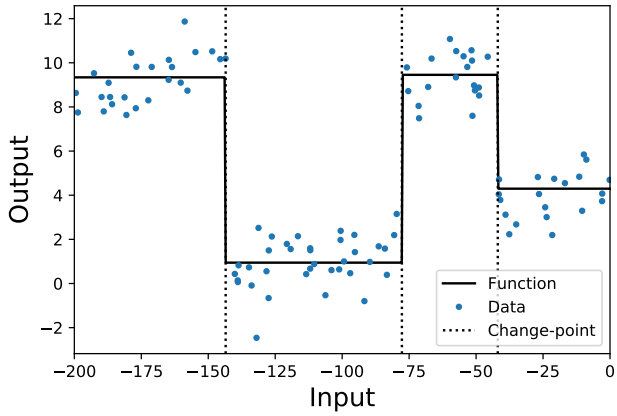


Figure 4.6: Piecewise-constant multiple-change-point function (MCP) and its 100 noisy observations.

provided by the Cabinet Office, Government of Japan. The dataset is a 450 m mesh data of 840×780 cells, and we downsample it into 280×260 for limited computational resources, as shown in Figure 4.9(a). The two-dimensional location (longitude and latitude) is used as input and the depth is used as output.

4.4.2 Comparing Methods

To the best of our knowledge, there is no existing method that directly addresses the ACPD problems. Hence, we compare the proposed meta-algorithm with two naive baselines: a random search and ϵ-greedy search. They explore the change-points using change scores computed by a given change-point detection algorithm. The random search samples the next input from a uniform distribution. ϵ-greedy search switches between the random search with probability ϵ and a greedy search with probability $1 - \epsilon$, where the greedy search suggests the middle point between the input with the greatest change score x^{1st} and the input with the second greatest change score x^{2st}: $x^{\text{next}} = (x^{\text{1st}} + x^{\text{2st}})/2$. We test different $\epsilon \in \{0.1, 0.5, 0.9\}$.

4.4.3 Change Scores

We employ the maximum likelihood in Equation (4.2) as the change score for the one-change-point functions $\mathrm{PT}_{\mathrm{bias}}, \mathrm{PT}_{\mathrm{slope}}$, NI, MO, and the multiple-change-point function, MCP. We use a third-order polynomial function for NI, a constant function for MCP, and linear functions for the others. We calibrate the original likelihood change scores $\{(x_i, s_i)\}_{i=1}^n$ to $\{((x_{i-1} + x_i)/2, s_i)\}_{i=2}^n$ to make them symmetric in the input space.

For computing the change score in the multiple-input problem, we use a spatial anomaly detection method AvgDiff [40]. It computes the change score of a data point (x, y) by comparing output y with outputs of its spatial nearest neighbors $NN(x)$. The score is defined as

$$s(x, y) = \sum_{(x_i, y_i) \in NN(x)} w(x, x_i) \left| y - y_i \right|,$$

where $w(x, x_i)$ denotes a spatial weight between x and x_i. Here, we do not explicitly assume a regression model of the target function like the maximum likelihood. Since AvgDiff uses spatial nearest neighbors, we think of it as implicitly assuming a regression model based on a k-nearest neighbor method that is a nonparametric model. We use the normalized inverse Euclidean distance

$$w(x, x_i) = \frac{1}{W_0 \|x - x_i\|},$$

where $W_0 = \sum_{(x_i, y_i) \in NN(x)} 1/\|x - x_i\|$, and the 10-nearest neighbors in the Euclidean distance

$$NN(x) = \operatorname*{arg\,min}_{A \subset \mathcal{D}, |A| = 10} \sum_{(x_i, y_i) \in A} \|x - x_i\|.$$

The choice of the change-point detection method or the change score definition depends on the types of change-points that a user wants to detect. A thorough comparison of the different methods for each function is beyond the scope of this chapter.

4.4.4 Change-Point Estimation and Evaluation Metrics

For one-change-point functions, we assume that the number of change-points is known and that the change-point is estimated by Equation (4.1). We evaluate the absolute error $|x^{\mathrm{cp}} - \hat{x}_t^{\mathrm{cp}}|$ between the estimated change-point $\hat{x}_t^{\mathrm{cp}}$ in iteration t and the ground-truth change-point x^{cp}.

For the multiple-change-point function MCP, we assume that the number of change-points is unknown. Change-points $\{\hat{x}_i^{\mathrm{cp}}\}_{i=1}^{\hat{n}}$ are estimated using a segmentation-based approach [18] with the Bayesian information criterion penalty [85] described as

$$\Omega(\{x_i^{\mathrm{cp}}\}_{i=1}^{\hat{n}}) = \frac{\hat{n}}{2} \log N.$$

The estimation error is measured by the Hausdorff distance [74]

$$\begin{aligned} &e(\{x_i^{\mathrm{cp}}\}_{i=1}^{n}, \{\hat{x}_i^{\mathrm{cp}}\}_{i=1}^{\hat{n}}) \\ &= \max\{ \max_{x^{\mathrm{cp}} \in \{x_i^{\mathrm{cp}}\}_{i=1}^{n}} \min_{\hat{x}^{\mathrm{cp}} \in \{\hat{x}_i^{\mathrm{cp}}\}_{i=1}^{\hat{n}}} |x^{\mathrm{cp}} - \hat{x}^{\mathrm{cp}}|, \max_{\hat{x}^{\mathrm{cp}} \in \{\hat{x}_i^{\mathrm{cp}}\}_{i=1}^{\hat{n}}} \min_{x^{\mathrm{cp}} \in \{x_i^{\mathrm{cp}}\}_{i=1}^{n}} |x^{\mathrm{cp}} - \hat{x}^{\mathrm{cp}}| \}. \end{aligned}$$

For the multiple-input function, i.e., the seafloor depth dataset, we consider inputs with rapid depth changes as change-points; however, there are no ground-truths. According to the motivation of ACPD, we use the change-points estimated using all data as the ground-truth change-points. In particular, we compute the ground-truth change scores using all $72,800$ data points (Figure 4.9(b)). We define inputs with the top k change scores and others as the ground-truth change-points and non-change-points, respectively. Each method is evaluated by the precision@k of change scores estimated by the posterior mean of the GP at the same inputs as the ground-truths. Because the number of change-points is assumed to be few compared to non-change-points, we set $k \in \{728(1\%), 3640(5\%), 7280(10\%), 10920(15\%), 14560\ (20\%)\}$.

We sample five points from a uniform distribution as the initial data for the one-dimensional input functions, and 20 points for the seafloor depth data. Each method

explores for $B = 100$ iterations. We measure each of the above performance values at each iteration and evaluate the mean value over B iterations and a final value after B iterations. We conduct each experiment 30 times.

4.4.5 Settings of Proposed Framework

The proposed method is a meta-algorithm, and it needs the specifications of the underlying BO method, i.e., covariance function k and acquisition function a. We use the Matérn 5/2 kernel (M52), where $\nu = 5/2$ in Equation (2.5). To study how to balance the trade-off between exploration and exploitation, we compare two acquisition functions: GP-UCB in Equation (2.10) and EI in Equation (2.11). We set the hyperparameter of the GP-UCB algorithm as $\beta_t^{1/2} \in \{0, 3, 6, 9, \infty\}$, where $\beta_t^{1/2} = 0$ and $\beta_t^{1/2} = \infty$ corresponds to a full exploitation strategy and a full exploration strategy, respectively. The hyperparameters of the covariance function are optimized by the type II maximum likelihood estimation every time a new observation is obtained.

4.4.6 Results and Discussions

Tables 4.1 and 4.2 summarize the results of the mean error over iterations and the error after the final iteration, respectively, for target functions besides the seafloor depth data. For $\mathrm{PT}_{\mathrm{bias}}$, $\mathrm{PT}_{\mathrm{slope}}$, and NI, the proposed meta-algorithm with EI and GP-UCB ($\beta_t^{1/2} = 3, 6, 9$) performed well. All results of EI except for the MCP mean error over iterations are better than those of the comparing methods. Through the iterations, the proposed methods effectively decreases the errors in Figure 4.7 and increases the precision@k in Figure 4.8. In contrast, the greedy approaches (GP-UCB ($\beta_t^{1/2} = 0$) and ϵ-greedy search) showed poor results. This might be because the greedy approaches fell into local optimums. As discussed in Section 4.3, change scores may vary depending on data. When the computed change scores at the non-change-points were incorrectly high, the greedy approaches may have fallen into incorrect solutions. The full exploitation strategy

without any exploration, GP-UCB ($\beta_t^{1/2} = 0$) did not improve after falling into the incorrect solution as shown in Figures 4.7 and 4.8. The proposed meta-algorithm not only exploited the knowledge, but also explored the input space uniformly in the early stages, which could provide the change scores similar to the ground-truth change scores. Owing to this property, the proposed meta-algorithm could avoid falling into incorrect solutions and thereby demonstrate the good performance.

For MCP, the greedy approaches (GP-UCB ($\beta_t^{1/2} = 0$) and ϵ-greedy search ($\epsilon = 0.1$)) showed worse results than the exploration approaches (Tables 4.1 and 4.2). This may be because it is required to explore uniformly the input domain for the three change-points distributed over the input domain. The proposed meta-algorithm with GP-UCB ($\beta_t^{1/2} \geq 6$), which prioritizes exploration over exploitation, demonstrated the performance equal to or slightly better than the comparing methods. This can be derived from the inherently difficult multiple change-point detection with an unknown number of change-points, for which the errors of the methods were not significantly different. The proposed meta-algorithm with GP-UCB ($\beta_t^{1/2} \geq 6$) exploited the knowledge while mainly focusing on exploring, which may provide the slightly better performance than the comparing methods for errors after the final iteration.

For the seafloor depth data, Tables 4.3 and 4.4 show the mean precision@k over the iterations and the precision@k after the final iteration, respectively. The proposed methods, except for the full exploitation strategy with GP-UCB ($\beta_t^{1/2} = 0$), performed better than the comparing methods for all different k. EI and GP-UCB ($\beta_t^{1/2} = 9$) worked well for the mean precision@k over iterations and precision@k after the final iteration, respectively. This may be because balancing between exploration and exploitation was important, compared to the one-dimensional-input functions for the large search space of the seafloor depth data. GP-UCB ($\beta_t^{1/2} = \infty$) implemented only exploration and no exploitation, which does not result in sample-efficient change-point detection. Figure 4.9(c,d,e,f) shows the observed data and the posterior means obtained by the GP of the proposed framework with EI, GP-UCB ($\beta_t^{1/2} = 9$), ϵ-greedy ($\epsilon = 0.1$) search, and the random search. The

Table 4.1: Means and standard deviations of the mean errors over iterations for 30 trials.

Method	$\mathrm{PT}_{\mathrm{bias}}$	$\mathrm{PT}_{\mathrm{slope}}$	NI	MO	MCP
Meta-ACPD (EI)	$\mathbf{23.6 \pm 20.1}$	23.6 ± 8.8	0.145 ± 0.117	15.4 ± 13.6	33.8 ± 6.7
Meta-ACPD (GP-UCB $\beta_t^{1/2} = 0$)	37.7 ± 29.0	35.9 ± 25.9	0.760 ± 0.657	24.5 ± 26.9	60.6 ± 21.2
Meta-ACPD (GP-UCB $\beta_t^{1/2} = 3$)	27.5 ± 20.8	$\mathbf{21.5 \pm 9.1}$	$\mathbf{0.123 \pm 0.082}$	16.6 ± 18.0	32.5 ± 4.2
Meta-ACPD (GP-UCB $\beta_t^{1/2} = 6$)	28.9 ± 18.4	26.0 ± 12.2	0.138 ± 0.091	$\mathbf{15.3 \pm 16.5}$	30.4 ± 2.8
Meta-ACPD (GP-UCB $\beta_t^{1/2} = 9$)	27.9 ± 23.8	25.5 ± 12.6	0.173 ± 0.126	17.0 ± 13.4	29.5 ± 1.9
Meta-ACPD (GP-UCB $\beta_t^{1/2} = \infty$)	45.2 ± 33.0	32.4 ± 11.6	0.242 ± 0.180	21.4 ± 8.8	$\mathbf{28.4 \pm 2.3}$
ϵ-greedy ($\epsilon = 0.1$)	40.1 ± 24.9	37.1 ± 21.3	0.434 ± 0.395	32.5 ± 20.5	61.3 ± 15.2
ϵ-greedy ($\epsilon = 0.5$)	35.7 ± 22.9	32.4 ± 19.3	0.239 ± 0.231	18.7 ± 12.5	34.1 ± 5.4
ϵ-greedy ($\epsilon = 0.9$)	37.4 ± 24.3	34.9 ± 12.7	0.233 ± 0.166	15.9 ± 6.9	29.3 ± 2.6
Random	42.9 ± 22.6	34.7 ± 15.8	0.273 ± 0.201	16.1 ± 7.2	30.0 ± 3.5

Table 4.2: Means and standard deviations of the errors after the final iteration for 30 trials.

Method	$\mathrm{PT}_{\mathrm{bias}}$	$\mathrm{PT}_{\mathrm{slope}}$	NI	MO	MCP
Meta-ACPD (EI)	15.5 ± 34.9	17.0 ± 18.3	0.0026 ± 0.0030	$\mathbf{0.1 \pm 0.1}$	12.6 ± 7.3
Meta-ACPD (GP-UCB $\beta_t^{1/2} = 0$)	36.4 ± 29.5	36.5 ± 26.5	0.6875 ± 0.6777	11.7 ± 26.3	36.8 ± 27.6
Meta-ACPD (GP-UCB $\beta_t^{1/2} = 3$)	$\mathbf{13.3 \pm 24.2}$	$\mathbf{13.1 \pm 12.0}$	$\mathbf{0.0008 \pm 0.0009}$	4.7 ± 18.0	13.5 ± 7.0
Meta-ACPD (GP-UCB $\beta_t^{1/2} = 6$)	16.7 ± 31.2	16.9 ± 14.6	0.0011 ± 0.0018	0.3 ± 1.4	$\mathbf{12.5 \pm 8.6}$
Meta-ACPD (GP-UCB $\beta_t^{1/2} = 9$)	18.7 ± 34.5	15.8 ± 12.2	0.0014 ± 0.0015	1.4 ± 6.1	12.8 ± 5.6
Meta-ACPD (GP-UCB $\beta_t^{1/2} = \infty$)	32.1 ± 45.7	22.2 ± 15.5	0.0410 ± 0.0596	0.9 ± 0.8	$\mathbf{12.5 \pm 8.8}$
ϵ-greedy ($\epsilon = 0.1$)	33.7 ± 32.0	33.3 ± 24.8	0.1751 ± 0.3558	15.1 ± 22.0	34.8 ± 21.9
ϵ-greedy ($\epsilon = 0.5$)	24.7 ± 37.7	22.2 ± 19.4	0.0343 ± 0.1200	0.7 ± 1.8	17.8 ± 6.9
ϵ-greedy ($\epsilon = 0.9$)	20.4 ± 33.7	22.6 ± 12.0	0.0177 ± 0.0151	0.6 ± 0.8	13.0 ± 6.1
Random	36.6 ± 43.4	24.2 ± 24.0	0.0268 ± 0.0398	0.9 ± 0.7	12.9 ± 7.0

proposed methods explored the ground-truth change-points and non-change-points in a balanced manner. Consequently, their change scores computed using only 120 data points correspond well to the ground-truth change scores (Figure 4.9(b)) computed using $72,800$ data points. In contrast, ϵ-greedy ($\epsilon = 0.1$) search (Figure 4.9(e)) only explored the few change-points and failed to explore the other change-points. The change scores by the random search (Figure 4.9(f)) do not correspond to the ground-truths clearly in the right bottom part.

Table 4.3: Means and standard deviations of mean precision@k of each method over iterations for 30 trials, where k is indicated by the percentage of the number of data.

Method	Precision@1%	Precision@5%	Precision@10%	Precision@15%	Precision@20%
Meta-ACPD (EI)	$\mathbf{0.145 \pm 0.072}$	0.321 ± 0.057	0.446 ± 0.050	0.497 ± 0.034	0.543 ± 0.027
Meta-ACPD (GP-UCB $\beta_t^{1/2} = 0$)	0.056 ± 0.054	0.193 ± 0.049	0.291 ± 0.052	0.347 ± 0.048	0.398 ± 0.052
Meta-ACPD (GP-UCB $\beta_t^{1/2} = 3$)	0.137 ± 0.051	0.331 ± 0.037	0.456 ± 0.038	0.502 ± 0.032	0.546 ± 0.026
Meta-ACPD (GP-UCB $\beta_t^{1/2} = 6$)	0.116 ± 0.061	0.328 ± 0.036	0.464 ± 0.032	0.510 ± 0.028	0.557 ± 0.027
Meta-ACPD (GP-UCB $\beta_t^{1/2} = 9$)	0.106 ± 0.050	$\mathbf{0.337 \pm 0.037}$	$\mathbf{0.472 \pm 0.037}$	$\mathbf{0.515 \pm 0.035}$	$\mathbf{0.561 \pm 0.031}$
Meta-ACPD (GP-UCB $\beta_t^{1/2} = \infty$)	0.039 ± 0.030	0.292 ± 0.053	0.420 ± 0.054	0.472 ± 0.046	0.538 ± 0.040
ϵ-greedy ($\epsilon = 0.1$)	0.044 ± 0.080	0.163 ± 0.063	0.259 ± 0.052	0.327 ± 0.046	0.388 ± 0.048
ϵ-greedy ($\epsilon = 0.5$)	0.038 ± 0.025	0.204 ± 0.052	0.329 ± 0.057	0.394 ± 0.047	0.453 ± 0.042
ϵ-greedy ($\epsilon = 0.9$)	0.042 ± 0.046	0.232 ± 0.061	0.374 ± 0.062	0.441 ± 0.051	0.504 ± 0.048
Random	0.040 ± 0.067	0.219 ± 0.059	0.355 ± 0.064	0.432 ± 0.062	0.505 ± 0.062

Table 4.4: Means and standard deviations of precision@k of each method after iterations for 30 trials, where k is indicated by the percentage of the number of data.

Method	Precision@1%	Precision@5%	Precision@10%	Precision@15%	Precision@20%
Meta-ACPD (EI)	$\mathbf{0.269 \pm 0.175}$	$\mathbf{0.464 \pm 0.062}$	$\mathbf{0.584 \pm 0.052}$	$\mathbf{0.613 \pm 0.041}$	0.638 ± 0.031
Meta-ACPD (GP-UCB $\beta_t^{1/2} = 0$)	0.084 ± 0.119	0.232 ± 0.077	0.324 ± 0.083	0.368 ± 0.089	0.393 ± 0.090
Meta-ACPD (GP-UCB $\beta_t^{1/2} = 3$)	0.260 ± 0.113	0.409 ± 0.072	0.542 ± 0.065	0.587 ± 0.051	0.620 ± 0.041
Meta-ACPD (GP-UCB $\beta_t^{1/2} = 6$)	0.207 ± 0.159	0.425 ± 0.081	0.553 ± 0.056	0.593 ± 0.042	0.632 ± 0.030
Meta-ACPD (GP-UCB $\beta_t^{1/2} = 9$)	0.192 ± 0.117	0.454 ± 0.059	0.580 ± 0.054	0.607 ± 0.037	0.637 ± 0.030
Meta-ACPD (GP-UCB $\beta_t^{1/2} = \infty$)	0.074 ± 0.091	0.426 ± 0.055	0.575 ± 0.048	0.593 ± 0.033	$\mathbf{0.640 \pm 0.027}$
ϵ-greedy ($\epsilon = 0.1$)	0.052 ± 0.082	0.184 ± 0.066	0.291 ± 0.053	0.362 ± 0.052	0.417 ± 0.053
ϵ-greedy ($\epsilon = 0.5$)	0.047 ± 0.048	0.242 ± 0.080	0.382 ± 0.074	0.450 ± 0.058	0.507 ± 0.047
ϵ-greedy ($\epsilon = 0.9$)	0.064 ± 0.112	0.284 ± 0.091	0.447 ± 0.069	0.516 ± 0.050	0.574 ± 0.043
Random	0.064 ± 0.115	0.307 ± 0.072	0.473 ± 0.078	0.541 ± 0.061	0.606 ± 0.047

4.5 Related Work

To the best of our knowledge, there is no work directly comparable to ACPD, and threfore, we review related works in a wide context by illustrating differences of their problem settings.

Change-point detection [7, 25] is the task of finding abrupt changes in time-series data. There are several closely related tasks such as concept drift [19], event detection [24], and time-series segmentation [38, 18]. Spatial outlier detection [40] is the task of detecting areas significantly different from the other areas in spatial data. Spatial outlier detection and image segmentation [56] can be considered multiple-input change-point detection problems. In general, change-point detection assumes either that each observation in a

time-series data is given at every time step in an online manner or that all parts of the time-series are given at once in a batch manner. In both the settings, the costs of acquiring data are not usually considered and their goals are just to provide the label of "change" or "normal" to each data point. In the ACPD problem, it is assumed that target data is not a time-series data. ACPD focuses on changes in a function [21] and the data is acquired in an active manner by paying some cost of measurement.

Further, ACPD is related to experimental design [11], where statistical models are used to determine efficient and effective series of experimental designs. BO [51, 67] suggests the next input to be evaluated based on past observations to seek the optimum of a black-box function that is expensive to evaluate. GPs [60] are the typical choice for models of black-box functions because they can handle uncertainty in the model. Various types of acquisition functions have been proposed, and they determine the next input by balancing exploration and exploitation based on the posterior mean and variance of the GP. We use the BO technique as the key component of our proposed framework because our objective is to find the optimum in a black-box change score function.

Active learning [65] is the task of learning a classifier model in an interactive manner by determining the next data point to query an oracle (e.g., a human annotator) for its label. The goal is to build a classifier that minimizes the predictive classification loss by choosing informative data points. Further, it can be considered that the problem setting of active learning is similar to that of BO, where the oracle and the label correspond to the black-box function and the real-valued output, respectively. However, their objectives are different: the predictive classification loss and the black-box function. The idea of active learning is applied to anomaly detection [15], where a data point is queried to be labeled an anomaly or a normal.

The closely related work to ACPD is interactive image segmentation [79, 81]. In this setting, all pixel values of an image are given, and at each iteration, a set of pixels is queried to an oracle for its label to segment the image. The problem may be regarded as an ACPD problem, where change-points are the boundaries of the segmentation, the input

is a two-dimensional coordinate of an image, and the output is a value and label of a pixel; however, all pixel values as a part of the output are provided. Further, ACPD mainly focuses on real-valued functions but not functions whose output is binary; however, it can handle them by properly selecting a change-point detection algorithm.

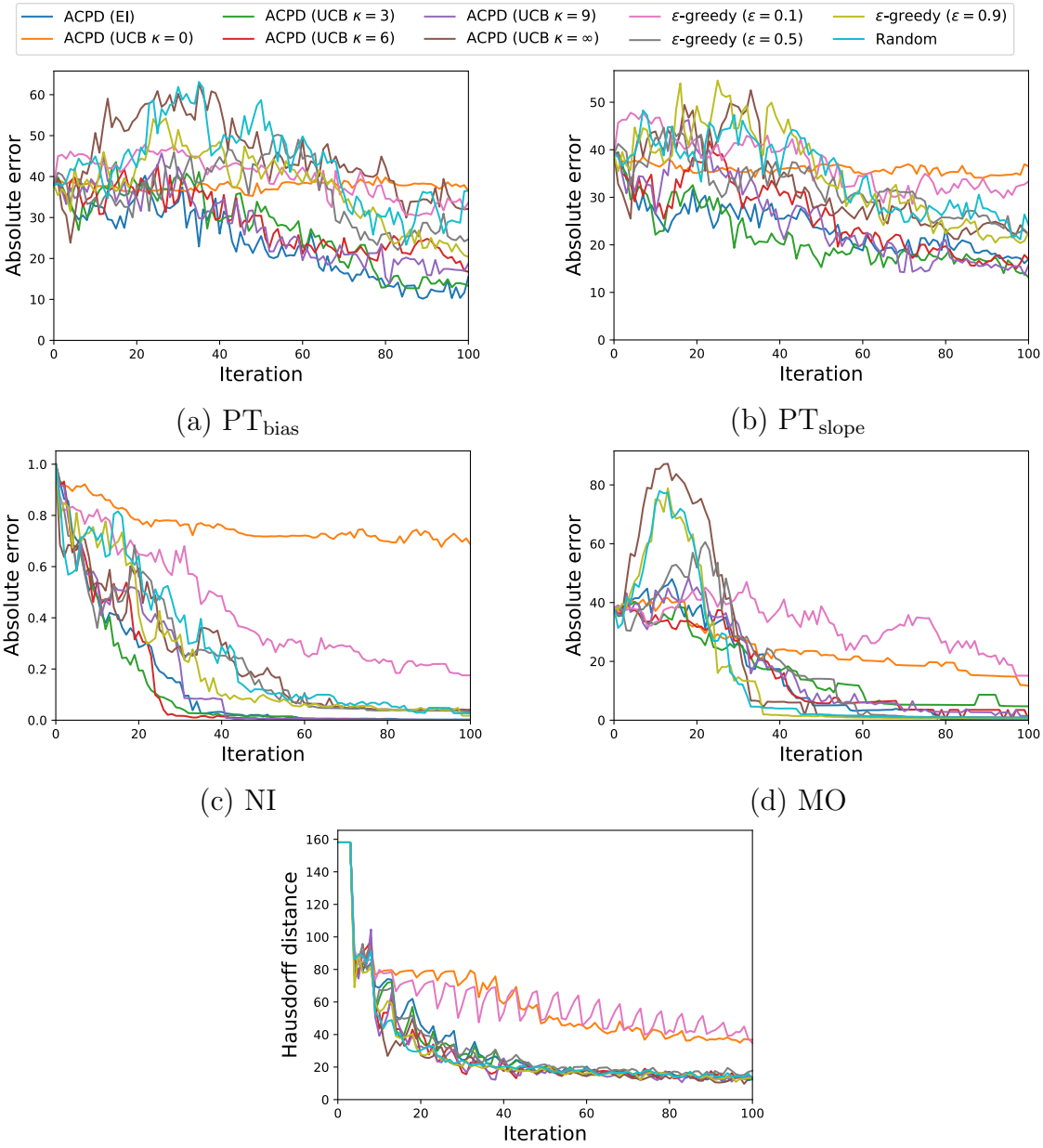
The other related work is active learning with drifting streaming data [80]. At each iteration, a learner obtains a piece of data, and she determines if she queries an oracle for its class label to train a classifier for drifting streaming data. This problem is similar to ACPD, where the input is time and the output is the data and its label. However, the difference is that data as a part of output is given and input corresponds the time.

4.6 Conclusion

Change-point detection problems in experimental data suffer from the small data problem because of the high costs to acquire data. The key is to select data with high information content for change point detection within the limited budget. We proposed ACPD, a novel active learning problem for cost-efficient change-point detection in an expensive-to-evaluate black-box function. In ACPD, a learner determines the next input to be evaluated to find the change-point in the function with as few evaluations as possible. We proposed the general meta-algorithm applicable to different types of data and change-points. The proposed meta-algorithm reuses an existing change-point detection method and a BO technique. The idea behind the proposed meta-algorithm is that we regard the ACPD problem as a black-box optimization problem of a change score function. The experimental results using the synthetic data and the real-world data showed that the proposed meta-algorithm outperformed the comparing methods considering exploration and exploitation for the different types of data and change-points.

In the future, we plan to extend the proposed meta-algorithm to the case where a user wants to detect change-points of N_c change definitions in the function using N_c change scores. In this case, it is considered possible to extend the proposed meta-algorithm by

modeling the relationship between input and N_c-dimensional change score with a multi-dimensional output GP [72]. Further, there may be more than one candidate regression model for the black-box function. For example, when a simple linear model and a complex nonlinear model are used, their model selection problems need to be addressed.



(a) $\mathrm{PT}_{\mathrm{bias}}$

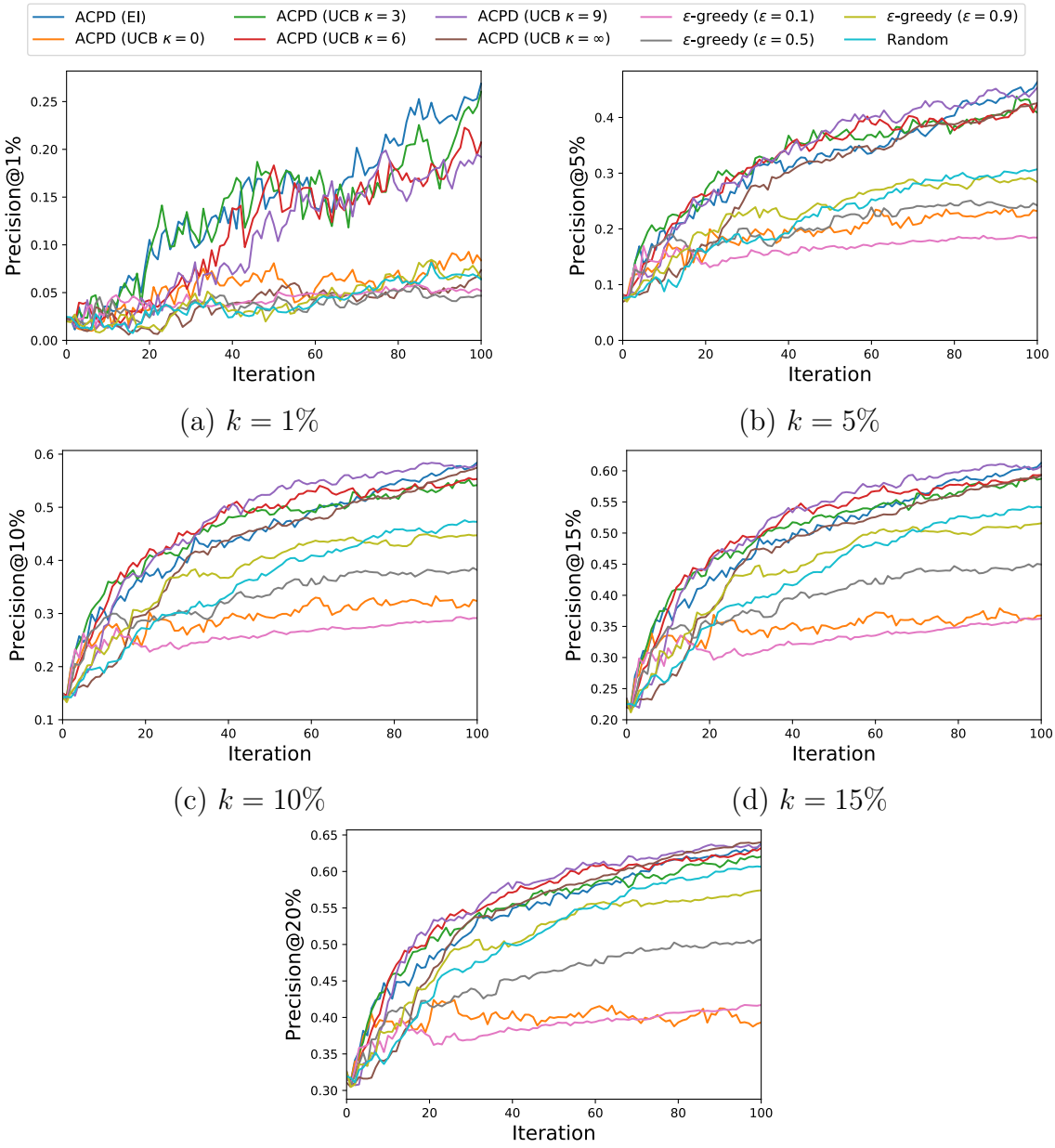
(b) $\mathrm{PT}_{\mathrm{slope}}$

(c) NI

(d) MO

(e) MCP

Figure 4.7: Error curves of the methods for the target functions.



(a) $k = 1\%$

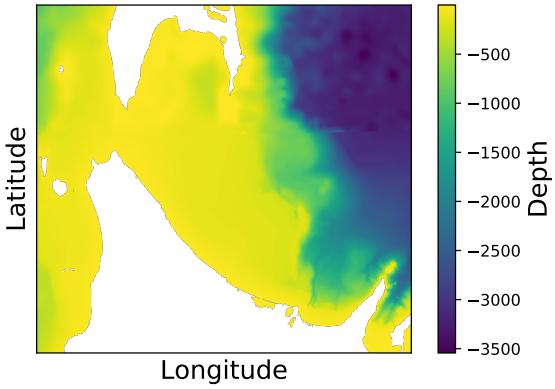
(b) $k = 5\%$

(c) $k = 10\%$

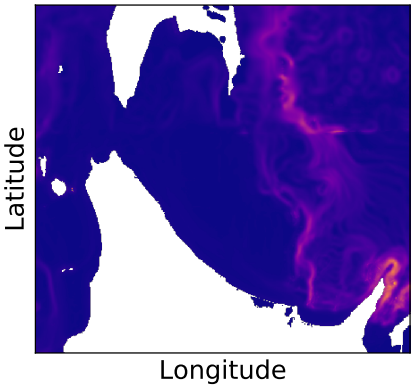
(d) $k = 15\%$

(e) $k = 20\%$

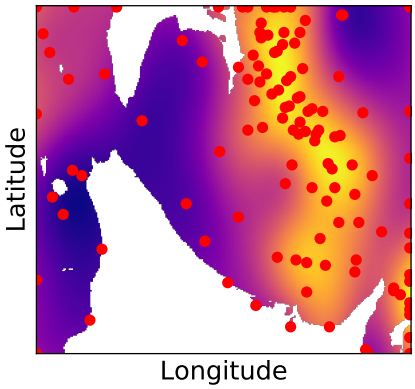
Figure 4.8: Precision@k curves of the methods for the seafloor depth.



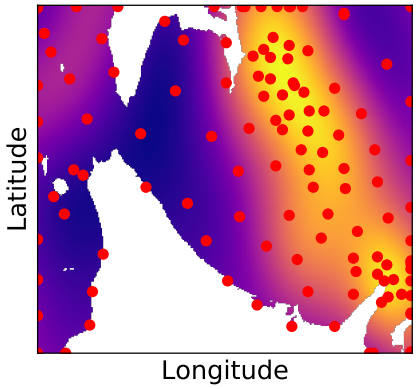
(a) Seafloor depth data



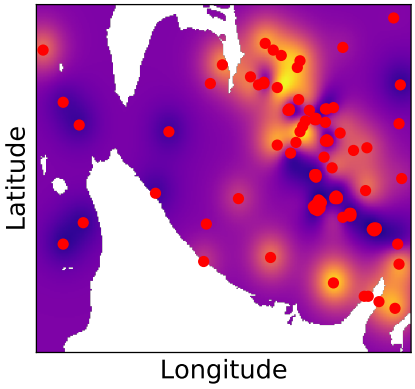
(b) Ground-truth



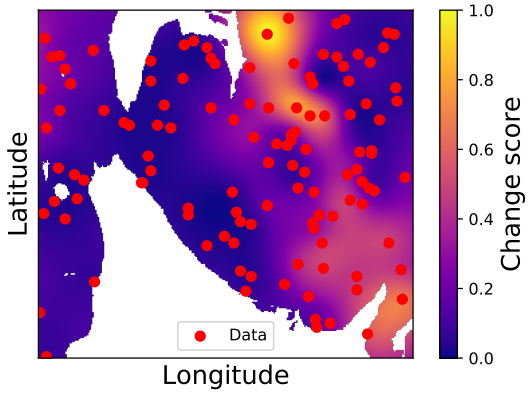
(c) Meta-ACPD (EI)



(d) Meta-ACPD (GP-UCB $\beta_t^{1/2} = 9$)



(e) ϵ-greedy ($\epsilon = 0.1$)



(f) Random

Figure 4.9: (a) Seafloor depth data. (b) Ground-truth change scores. (c),(d),(e),(f) 120 data points and change scores estimated by a GP using Meta-ACPD with EI and GP-UCB ($\beta_t^{1/2} = 9$), ϵ-greedy search ($\epsilon = 0.1$), and random search.

Chapter 5

Bayesian Optimization with Partially Specified Queries

The additional data settings including ACPD in Chapter 4 often handle deterministic input queries, that is, the learner can query a black-box function for its value at a fully specified input. However, the full specification of input queries requires large costs in some scenarios. This chapter considers a BO problem with a partial specification of input queries, which is potentially cheaper than the full specification. Thus, in contrast to deterministic fully specified queries, partially specified queries are random. Further, we propose algorithms that balance exploration and exploitation of information to detect the optimal solution based on posterior sampling and a multi-armed bandit approach.

5.1 Introduction

BO [66] is a promising approach for black-box optimization of an expensive-to-evaluate unknown function. Its applications include the hyperparameter tuning of machine learning models with the highest predictive accuracy [67] and searching compound structures with desirable properties [39]. BO is conducted in an iterative manner. In each iteration, a BO method determines the values of input variables in an input domain $\mathcal{X} \subset \mathbb{R}^d$, evaluates a

black-box function $f : \mathcal{X} \to \mathbb{R}$ at that input query and observes the corresponding output $y \in \mathbb{R}$. The final goal is to find the input that maximizes the function with as few function evaluations as possible.

Whereas a learner can fully specify all values of input variables in the standard BO setting, this is not true for all cases. For example, consider the design of airfoils with low self-noise. In this case, we ask a manufacturer to produce an airfoil by specifying its various features, including angle of attack and chord length. However, it is costly to specify all feature values; instead, with a limited budget, we are restricted to specify only a subset of the features. After obtaining the ordered airfoil, the values of the unspecified features are revealed, and its self-noise is measured via an acoustic test.

Another example is crowdsourcing, by which human-intelligent tasks are outsourced to an undetermined number of people on the Internet. Most existing crowdsourcing platforms allow us to ask workers to do the task by specifying conditions such as age, country, and experience in order to obtain high quality deliverables [46]. However, if the conditions are too strict, there is a risk that there will be no workers available to participate in the task. Upon the completion of the task, we observe the quality of the deliverable and the features of the workers who worked on the task.

In this chapter, we propose a novel Gaussian process bandit framework, which we call BOPSQ. In BOPSQ, unlike the standard BO setting, a learner selects a *subset* of input variables and specifies their values. We call such an input query a *partially specified query*. Next, the learner observes the values of the unspecified input variables determined according to a known or unknown distribution. Then, the learner evaluates the black-box function at the full input variables to obtain the corresponding output. Hence, the learner is required to consider the extent to which the input variables contribute to the function's output as well as which values the unspecified input variables will take. Williams et al. [83], Lattimore et al. [43] also handled partially specified queries. However, in their works, the input variables whose values are specified are fixed, or infinite input spaces are not considered for the Gaussian process bandit.

We propose two algorithms for cases of known and unknown input distributions. In the case of the known input distribution, the uncertainty lies only in the function estimation, and we can naturally extend a posterior sampling approach to this setting. In the case of the unknown input distribution, the learner further needs to take the uncertainty in the distribution estimation into account. We adopt a multi-armed bandit approach to appropriately explore the input distribution.

The remainder of this chapter is organized as follows. Section 5.2 describes notations used in this chapter and the problem setting of BOPFQ. Section 5.3 introduces two algorithms for the cases where the input distribution is known and unknown, providing their regret upper bounds. The proofs are given in Section 5.4. The empirical evaluation of the proposed algorithms are demonstrated in Section 5.5 using test functions and real-world functions. Section 5.6 summarizes the related works to BOPFQ. Section 5.7 gives concluding remarks with possible future directions.

5.2 Problem Setting

We propose a novel Gaussian process bandit problem that utilizes partially specified queries, BOPSQ. We first define some terms and notations. Then, we introduce the framework and define regret for BOPSQ.

5.2.1 Notation

Let $\mathcal{X} = \mathcal{X}^{(1)} \times \ldots \times \mathcal{X}^{(d)} \subset \mathbb{R}^d$, $d \in \mathbb{N}$, be a compact domain and $f : \mathcal{X} \to \mathbb{R}$ be an unknown function of interest. We call the indices of input variables which a learner can specify the values of, a *control variable set*, denoted by $I \subset [d]$. We call the remaining indices of input variables which the learner can not specify the values of, an *uncontrol variable set*, denoted by $\bar{I} = [d] \setminus I$. Let $\mathcal{I} \subset 2^{[d]}$ be a family of control variable sets. We call the values of control (uncontrol) variable set *control (uncontrol) variables*, denoted by $x^I \in \mathcal{X}^I$ ($x^{\bar{I}} \in \mathcal{X}^{\bar{I}}$). Here, $x^I = \{x^{(i)} \in \mathcal{X}^{(i)} \mid i \in I\}$ and $\mathcal{X}^I = \mathcal{X}^{i_1} \times \ldots \times \mathcal{X}^{i_k}$ for

$I = \{i_1, \ldots, i_k\} \subset [d]$. Let $X = (X^{(1)}, ..., X^{(d)})^\top \sim P(X)$ be a random vector according to an input distribution P over $\mathcal{X}$ and Y be a random variable over $\mathbb{R}$ corresponding to the output variable. With a slight abuse of notation, we write $f(x^I, x^{\bar{I}})$ for $f(x)$.

5.2.2 Framework

The optimization procedure consists of T iterations. A learner is given a non-empty family of control variable sets $\mathcal{I} \subset 2^{[d]}$, which we assume is static over T iterations for simplicity in this chapter. In each iteration, the learner selects a control variable set $I \in \mathcal{I}$ and specifies their values $x^I \in \mathcal{X}^I$. Next, the learner observes uncontrol variables $X^{\bar{I}}$ drawn from a conditional distribution $P(X^{\bar{I}} \mid X^I = x^I)$, which is either known or unknown. By evaluating a black-box function f at the input variables $(x^I, X^{\bar{I}})$, the learner observes the corresponding output $Y = f(x^I, X^{\bar{I}}) + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma^2)$.

In the standard BO setting, the goal is to find the optimal input variables x^*. However, in BOPSQ, the learner may not always select x^* and obtain $f(x^*)$ because the input query is random. Therefore, for the objective function, we consider the conditional expectation of $f(x^I, X^{\bar{I}})$ over $X^{\bar{I}}$ given $X^I = x^I$. We define the goal as finding the optimal control variable set $I^* \in \mathcal{I}$ and control variables $x^{I^*} \in \mathcal{X}^{I^*}$ in as few evaluations as possible, defined as

$$(I^*, x^{I^*}) = \underset{I \in \mathcal{I}, x^I \in \mathcal{X}^I}{\arg\max}\, \mathbb{E}[f(X^I, X^{\bar{I}}) \mid f, X^I = x^I]. \tag{5.1}$$

Hereafter, we write x^{I^*} instead of (I^*, x^{I^*}) for simplicity.

Some specific forms set as the family of control variable sets $\mathcal{I}$ recover existing BO settings. The simple setting in which the learner can select all input variables as a control variable set, $\mathcal{I} = \{[d]\}$, corresponds to the standard BO setting. When the family of control variable sets is always fixed to one particular subset, $\mathcal{I} = \{I\}, I \subset [d], 0 < |I| < d$, we consider this setting as BO with environmental variables [83]. In this sense, BOPSQ is a generalization of these BO settings. The major difference appears when the

learner is required to select a control variable set for $|\mathcal{I}| > 1$. Moreover, when the input distribution $P(X)$ is unknown, the learner is required to consider the uncertainty of the input distribution estimation as well as the function estimation.

Let $I \subset J \in \mathcal{I}$, that is, I be a subset of J. Then, it is easy to see that the maximum of the objective function for I in Equation (5.1) is no larger than that for J, i.e., $\max_{x^I \in \mathcal{X}^I} \mathbb{E}[f(X^I, X^{\bar{I}}) \mid f, X^I = x^I] \leq \max_{x^J \in \mathcal{X}^J} \mathbb{E}[f(X^J, X^{\bar{J}}) \mid f, X^J = x^J]$. Thus, we assume the family of control variable sets contains no control variable set that is a subset of other control variable sets.

5.2.3 Regret

The final goal is to find the optimal control variables x^{I^*} defined in Equation (5.1). We define instantaneous regret $\mathrm{IR}(t)$ in iteration t, which evaluates the gap between the optimal solution and the choice x^{I_t}, described as

$$\mathrm{IR}(t) = \mathbb{E}[f(X^{I^*}, X^{\bar{I^*}}) \mid f, X^{I^*} = x^{I^*}] - \mathbb{E}[f(X^{I_t}, X^{\bar{I_t}}) \mid f, X^{I_t} = x^{I_t}].$$

The resultant performance measure, Bayes cumulative regret $\mathrm{BayesRegret}(T)$, is the expectation and summation over T iterations of the instantaneous regret, defined as

$$\mathrm{BayesRegret}(T) = \sum_{t=1}^{T} \mathbb{E}\left[\mathrm{IR}(t)\right].$$

Our focus is designing algorithms whose regret is sublinear. The sublinear regret leads to *no-regret*, which means that the mean regret over T asymptotically approaches 0.

$$\lim_{T \to \infty} \frac{\mathrm{BayesRegret}(T)}{T} \to 0.$$

In other words, an algorithm that satisfies no-regret asymptotically finds the optimal solution.

5.3 Algorithms

We consider two cases where the input distribution is known and unknown, propose two algorithms based on posterior sampling for these cases, and also provide their regret bounds that are sublinear for popular kernels.

5.3.1 TSPSQ-KNOWN for Known Input Distribution

We first focus on the case where the joint input distribution $P(X)$ or the set of conditional distributions $\{P(X^{\bar{I}} \mid X^I)\}_{I \in \mathcal{I}}$ is known. For example, in the airfoil design provided by a manufacturer, this case corresponds to the situation in which the manufacturer publishes random options for the unspecified features. In the crowdsourcing example, this case corresponds to a situation where the past offer history is accessible. We propose an algorithm based on Thompson sampling, which we call Thompson Sampling with Partially Specified Queries for the known input distribution case (TSPSQ-KNOWN).

Acquisition Function

In iteration t, TSPSQ-KNOWN determines the next control variables as

$$x^{I_t} = \underset{I \in \mathcal{I}, x^I \in \mathcal{X}^I}{\arg\max}\, \mathbb{E}[g_t(X^I, X^{\bar{I}}) \mid g_t, X^I = x^I], \tag{5.2}$$

where g_t is a sample from the GP posterior $g_t \sim \mathcal{GP}(\mu_{t-1}, k_{t-1})$. The pseudo-code is given in Algorithm 4.

Equation (5.2) approaches the ground-truth objective in Equation (5.1) as the posterior sample g_t approximates the ground-truth function f better with more observations. TSPSQ-KNOWN is a natural extension of Thompson sampling; if we set $\mathcal{I} = \{[d]\}$ as the standard BO setting, the acquisition function in Equation (5.2) is reduced to $g_t(x)$.

Regret Bound

We first introduce the notion of maximum information gain, which is the basis of typical regret analysis for BO. The maximum information gain in iteration T, denoted by γ_T, is the maximum possible information gain achievable by any algorithm for f via queries $A = \{x_1, x_2, ..., x_T\} \subset \mathcal{X}$ and the corresponding outputs $y_A = \{y_1, y_2, ..., y_T\}$, defined as

$$\gamma_T = \max_{A \subset \mathcal{X}, |A|=T} I(f; y_A). \tag{5.3}$$

Here, $I(f; y_A)$ denotes the mutual information between f and y_A. Srinivas et al. [70] provided the bounds of γ_T for various types of kernels: $\gamma_T \in O(\log T)$ for the linear kernel $k(x, y) = x^\top y$, $\gamma_T \in O\left((\log T)^{d+1}\right)$ for the Gaussian kernel $k(x, y) = \exp\left(0.5\|x - y\|^2/l^2\right)$, and $\gamma_T \in O\left(T^{d(d+1)/(2\nu+d(d+1))} \log T\right)$ for the Matérn kernel $k(x, y) = \left(2^{1-\nu}/\Gamma(\nu)\right) r^\nu B_\nu(r)$, where $r = \left(\sqrt{2\nu}/l\right) \|x - y\|$, B_ν is the modified Bessel function, and $l, \nu > 0$.

We then derive the regret bound of TSPSQ-KNOWN for the case where the ground-truth input distribution is known.

Theorem 2 *Let $\mathcal{X} \subset [0, 1]^d$, $d \in \mathbb{N}$, be a compact domain. Assume $k(x, x') \leq 1$ and that there exist constants $a', b' > 0$ for the partial derivatives of sample paths f such that*

$$l > 0, \forall j \in [d],\ \mathbb{P}\left(\sup_{x \in \mathcal{X}} \left|\frac{\partial f(x)}{\partial x^{(j)}}\right| > l\right) \leq a' e^{-(l/b')^2}. \tag{5.4}$$

Then, by running TSPSQ-KNOWN *for T iterations, it holds that*

$$\text{BayesRegret}(T) \in O(\sqrt{dT\gamma_T \log T}).$$

The analysis is mainly based on the techniques by Russo and Roy [62], Srinivas et al. [70], where the cumulative regret is decomposed into terms with respect to discretization errors of $\mathcal{X}$, the difference between f and the upper confidence bound $\mu_{t-1}(x) + \beta_t^{1/2} \sigma_{t-1}(x)$, and the difference between x^{I^*} and x^{I_t}. A detailed proof of Theorem 2 is found in Sec-

Algorithm 4 TSPSQ-KNOWN (TSPSQ-UNKNOWN)

Input: A GP prior $\mathcal{GP}(\mathbf{0}, k)$, a query budget T, and a family of control variable sets $\mathcal{I}$ (and a set of conditional distributions $\{P(X^{\bar{I}} \mid X^I)\}_{I \in \mathcal{I}}$ for TSPSQ-KNOWN)
Initialize $\mathcal{D}_0 \leftarrow \emptyset$ (and $\mathcal{S}_0^{(i)} \leftarrow \emptyset$ for $i \in \bar{I}_t$ for TSPSQ-UNKNOWN)
for $t = 1, 2, \ldots, T$ **do**
  Determine the next control variables x^{I_t} using Equation (5.2) for TSPSQ-KNOWN (or Equation (5.6) for TSPSQ-UNKNOWN)
  Observe uncontrol variables $X^{\bar{I}_t}$
  Evaluate f at $(x^{I_t}, X^{\bar{I}_t})$, and observe the corresponding output y_t
  Update $\mathcal{D}_t \leftarrow \mathcal{D}_{t-1} \cup \{((x^{I_t}, X^{\bar{I}_t}), y_t)\}$ and $\mathcal{GP}$ (and $\mathcal{S}_t^{(i)} \leftarrow \mathcal{S}_{t-1}^{(i)} \cup \{X^{(i)}\}$ for $i \in \bar{I}_t$ for TSPSQ-UNKNOWN)
end for

tion 5.4.1.

Note that the growth rate of the regret bound in T matches the existing results of Thompson sampling and GP-UCB in the standard BO setting shown by Russo and Roy [62], Srinivas et al. [70], respectively, though their definitions of regret are different. This is because the ground-truth input distribution is given, and the uncertainty in estimation is therefore only in f, as in the standard BO setting. The regret bound is sublinear for the popular kernels, such as the linear kernel, the Gaussian kernel, and the Matén kernel.

5.3.2 TSPSQ-UNKNOWN for Unknown Input Distribution

Next, we consider the case where the joint input distribution $P(X)$, or the set of conditional distributions $\{P(X^{\bar{I}} \mid X^I)\}_{I \in \mathcal{I}}$ is unknown. We propose Thompson Sampling with Partially Specified Queries for the unknown input distribution (TSPSQ-UNKNOWN).

Unlike the known input distribution case, the estimation of the input distribution needs to be addressed in the unknown input distribution case. However, without any assumption on the model of distributions, this would require an exponential number of samples in d to obtain the estimate of the distribution with sufficiently small error, because it would be inevitable to estimate the conditional distribution for each combination of control variables (a discussion is presented in the following). Therefore, to make the estimation

statistically tractable, we assume that input variables are independent of each other; and the cumulative distribution function $F(X \leq x)$ is decomposed as

$$F(X \leq x) = \prod_{i=1}^{d} F^{(i)}(X^{(i)} \leq x^{(i)}).$$

Further, we assume F^i are continuous over $\mathcal{X}^{(i)}$ for $i \in [d]$. Based on the assumptions, we propose an algorithm and derive its regret upper bound.

Discussion on Estimating an Unknown Dependent Input Distribution

We consider a simple toy example that illustrates the essential hardness of the problem when the input distribution is not necessarily independent.

For simplicity we consider the binary input space, $\mathcal{X} = \{0, 1\}^6$ with $\mathcal{I} = \{I \subset [d] \mid |I| = 3\}$, but the discussion below is easily extended to the continuous case with general input dimension d and the number of control variables $m = |I|$. Let us consider the case that the learner already knows the complete information of the objective function $f(x)$, which is expressed as

$$f(x) = \begin{cases} 1, & \text{if } |\{i \mid x^{(i)} = 0\}| \geq 4, \\ 0, & \text{otherwise.} \end{cases}$$

Now, consider the following two distributions P and Q. Under P, $X^{(i)}$ is i.i.d. with $P(X^{(i)} = 0) = \epsilon$ for sufficiently small $\epsilon > 0$. Under Q, the sequences in $\mathcal{S}$ given below appear more frequently, satisfying

$$Q(x) = \begin{cases} P(x)/\epsilon, & \text{if } x \in \mathcal{S}, \\ P(x)/A, & \text{otherwise,} \end{cases}$$

where

$$\mathcal{S} = \{000011,\ 100011,\ 010011,\ 001011\},$$
$$A = \frac{1 - \sum_{x \in \mathcal{S}} P(x)}{1 - \sum_{x \in \mathcal{S}} Q(x)} = \frac{1 - \epsilon^3(1-\epsilon)^2(3-2\epsilon^2)}{1 - \epsilon^2(1-\epsilon)^2(3-2\epsilon^2)} \approx 1.$$

We can see that, under Q, it is uniquely optimal to choose $I = \{1, 2, 3\}$ with $x^I = 000$. This is because, in this choice $Q(X^{\bar{I}} = 011 | X^I = 000) \approx 1/2$ whereas $X^{\bar{I}}$ almost always becomes 111 (resulting in $f(x) = 0$) in any other choice. Nevertheless, it requires an enormous number of samples to distinguish P and Q unless choosing $I = \{1, 2, 3\}$ with $x = 000$, because $X^{\bar{I}}$ almost always takes value 111 in all other cases.

By the same argument, for arbitrary I such that $|I| = 3$, we can consider a distribution Q' such that choosing I with $x^I = 000$ is optimal but Q' cannot be distinguished from P unless trying this choice. Therefore, it is necessary to try combinatorially many candidates of I to find the input such that $f(x)$ becomes 1 with a nonnegligible probability. From this observation, we can conclude that dependent input distributions essentially make the task exponentially hard.

Acquisition Function

We first define a function $\bar{g}_t$ using a posterior sample $g_t \sim \mathcal{GP}(\mu_{t-1}, k_{t-1})$, written as

$$\bar{g}_t\left(x^I, x^{\bar{I}}\right) = g_t\left(x^I, x^{\bar{I}}\right) + \sum_{i \in \bar{I}} \frac{\alpha_t}{\sqrt{|\mathcal{S}_{t-1}^{(i)}|}}, \tag{5.5}$$

where $\mathcal{S}_{t-1}^{(i)}$ is a set of observations of the uncontrol variables until iteration $t-1$ for $i \in [d]$, defined as $\mathcal{S}_{t-1}^{(i)} = \{X_s^{(i)} \mid i \in \bar{I}_s, s \in [t-1]\}$, and α_t is a monotonically increasing function in t. TSPSQ-UNKNOWN determines the next control variables as

$$x^{I_t} = \underset{x^I \in \mathcal{X}^I, I \in \mathcal{I}}{\operatorname{argmax}} \mathbb{E}_{X^{\bar{I}} \sim \hat{F}^I}\left[\bar{g}_t\left(x^I, X^{\bar{I}}\right)\right], \tag{5.6}$$

where $\hat{F}^I$ denotes the empirical distribution function for I, defined as $\hat{F}^I(X^I \leq x^I) = \prod_{i \in I} \hat{F}^{(i)}(X^{(i)} \leq x^{(i)})$, $\hat{F}^{(i)}(X^{(i)} \leq x^{(i)}) = 1/|\mathcal{S}_{t-1}^{(i)}| \sum_{\hat{x}^{(i)} \in \mathcal{S}_{t-1}^{(i)}} \mathbb{1}[\hat{x}^{(i)} \leq x^{(i)}]$ The pseudo-code is given in Algorithm 4.

The acquisition function of TSPSQ-UNKNOWN in Equation (5.6) works similarly to that of TSPSQ-KNOWN in Equation (5.2) for the known input distribution case. TSPSQ-UNKNOWN takes expectation using the empirical distribution instead of the ground-truth distribution. The second term on the right hand side of Equation (5.5) encourages increasing the number of samples from the input distributions with few samples and prevents the learner from falling into a local solution caused by misestimation of the input distribution. Hence, α_t controls the exploration-exploitation trade-off in the distribution estimation. This is conceptually same as the approach of the UCB algorithms for the multi-armed bandit problems [6]. In this sense, we can see that BOPSQ is a mixture of the Gaussian process bandit problems and the multi-armed bandit problems.

TSPSQ-UNKNOWN avoids a regret bound exponentially dependent on the number of input variables d for the independence assumption, as shown later, because it need not to consider the combinations of d-dimensional domains in the distribution estimation. However, its computational cost in each iteration still exponentially depends on d due to the combinations of $\mathcal{S}_{t-1}^{(i)}$ for $i \in \bar{I}$. This cost could be a serious burden when d, or the size of the uncontrol variable set $|\bar{I}|$ is large. A general solution to this problem would not be easily found; however, the problem can be mitigated when some assumptions hold. For example, Kandasamy et al. [36], Mutny and Krause [53] assumed the additivity of f with respect to each $x^{(i)}$ for $i \in [d]$, i.e., $f(x) = \sum_{i \in [d]} f_i(x^{(i)})$, and then the computational cost is reduced to linear in d. Because the problem can be separated from BOPSQ, for simplicity, we do not consider this issue in depth.

Regret Bound

We then derive a regret bound for the unknown input distribution case.

Theorem 3 *Let $\mathcal{X} \subset [0,1]^d, d \in \mathbb{N}$, be a compact domain. Assume $k(x,x') \leq 1$ and that there exist constants $a, b, a', b' > 0$ for the sample paths f and their partial derivatives such that Assumption* (5.4) *holds and*

$$l > 0,\ \mathbb{P}\left(\sup_{x \in \mathcal{X}} |f(x)| > l\right) \leq a e^{-(l/b)^2}. \tag{5.7}$$

Suppose an unknown cumulative distribution function of the input variables is factorized into $F(X \leq x) = \prod_{i=1}^{d} F^{(i)}(X^{(i)} \leq x^{(i)})$, and $F^{(i)}$ is continuous over $\mathcal{X}^{(i)}$. Define $\alpha_t = 2b' \log t$. Then, by running TSPSQ-UNKNOWN *for T iterations, it holds that*

$$\mathrm{BayesRegret}(T) \in O\left(\sqrt{dT \log T}\left(\sqrt{\gamma_T} + \sqrt{m \log T}\right)\right),$$

where γ_T is the maximum information gain defined in Equation (5.3) *and $m = \max_{I \in \mathcal{I}} |\bar{I}|$ is the maximum number of dimensions of uncontrol variables.*

The main difference of the proof from that for Theorem 2 lies in the evaluation of the distribution estimation, which does not appear in the standard BO setting. Therefore, we employ techniques in multi-armed bandit problems and the Dvoretzky–Kiefer–Wolfowitz inequality [16, 50] to derive the upper bound. The whole proof is found in Section 5.4.2.

The growth rate of the regret bound in T in Theorem 3 matches the existing results for the standard BO setting [62, 70], as with TSPSQ-KNOWN. This is because the bound for the estimation of f is still dominant in the total regret bound compared to a bound for the estimation of the input distribution thanks to the independence assumption. Note that the bound depends only on the square root of d, and no exponential term appears in d. As the maximum number of dimensions of uncontrol variables m decreases, the bound improves. This is because the learner can specify the values of the many input variables as desired and the error with respect to the distribution estimation becomes small. Note that the growth rate in T of the maximum information gain γ_T is greater than $m \log T$ for the linear, Gaussian, and Matérn kernels, as shown in Section 5.3.1.

5.4 Proofs

5.4.1 Proof of Theorem 2

In this section, we present the proof of Theorem 2, where the input distribution is known.

We first introduce a relevant result. Srinivas et al. [70] provided a bound of sum of posterior variances over T iterations using the maximum information gain γ_T as

$$\sum_{t=1}^{T} \sigma_{t-1}^2(x_t) \leq \frac{2}{\log(1+\sigma^{-2})}\gamma_T. \tag{5.8}$$

The analysis basically follows the frameworks by Russo and Roy [62], Srinivas et al. [70]. First, we discretize the domain $\mathcal{X}$ in iteration t into grid points $[\mathcal{X}]_t$ so that $|[\mathcal{X}]_t| = \tau_t^d$ and $\|x - [x]_t\|_1 \leq d/\tau_t$ for all $x \in \mathcal{X}$, where $[x]_t$ is the closest point in $[\mathcal{X}]_t$ and $\tau_t = t^2 da'b'\sqrt{\pi}$. Note that this discretization is deterministic.

Define an upper confidence bound (UCB) sequence $U_t(\cdot) = \mu_{t-1}(\cdot) + \beta_t^{1/2}\sigma_{t-1}(\cdot)$ in iteration t, where $\beta_t = 4(d+1)\log(t) + 2d\log(da'b'\sqrt{\pi})$. Using the UCB sequence, we decompose BayesRegret(T) into

$$\begin{aligned}
\text{BayesRegret}(T) = &\underbrace{\sum_{t=1}^{T} \mathbb{E}\left[f(X^{I^*}, X^{\bar{I}^*}) - f([X^{I^*}]_t, [X^{\bar{I}^*}]_t)\right]}_{A_1} \\
&+ \underbrace{\sum_{t=1}^{T} \mathbb{E}\left[f([X^{I^*}]_t, [X^{\bar{I}^*}]_t) - U_t([X^{I^*}]_t, [X^{\bar{I}^*}]_t)\right]}_{A_2} \\
&+ \underbrace{\sum_{t=1}^{T} \mathbb{E}\Big[U_t([X^{I^*}]_t, [X^{\bar{I}^*}]_t) - U_t([X^{I_t}]_t, [X^{\bar{I}_t}])\Big]}_{A_3} \\
&+ \underbrace{\sum_{t=1}^{T} \mathbb{E}\Big[U_t([X^{I_t}]_t, [X^{\bar{I}_t}]_t) - f([X^{I_t}]_t, [X^{\bar{I}_t}]_t)\Big]}_{A_4}
\end{aligned}$$

$$+ \underbrace{\sum_{t=1}^{T} \mathbb{E}\left[f([X^{I_t}]_t, [X^{\bar{I}_t}]_t) - f(X^{I_t}, X^{\bar{I}_t}) \right]}_{A_5}. \tag{5.9}$$

We will bound the terms A_1 and A_5. By Assumption 5.4, we have

$$\begin{aligned}
A_1 &\leq \sum_{t=1}^{T} \mathbb{E}\left[\frac{d}{\tau_t} \sup_{x \in \mathcal{X}, j \in [d]} \left| \frac{\partial f(x)}{\partial x^{(j)}} \right| \right] \\
&\leq d \sum_{t=1}^{T} \frac{1}{t^2 da'b'\sqrt{\pi}} \int_0^\infty \mathbb{P}\left(\sup_{x \in \mathcal{X}, j \in [d]} \left| \frac{\partial f(x)}{\partial x^{(j)}} \right| \geq l \right) dl \\
&\leq \frac{\sqrt{\pi^3}}{6a'b'} \int_0^\infty a' e^{-l^2/b'^2} dl \\
&= \frac{\pi^2}{12}.
\end{aligned} \tag{5.10}$$

By applying the same argument to A_5 as above, we obtain the same upper-bound.

Next, we will upper-bound A_2. For a random variable $Z \sim \mathcal{N}(\mu, \eta^2)$ with $\mu \leq 0$, $\mathbb{E}[Z \mathbb{1}[Z > 0]] = \int_0^\infty z/(\eta\sqrt{2\pi}) \exp(-(z-\mu)^2/(2\eta^2)) dz \leq (\eta/\sqrt{2\pi}) \exp\left(-\mu^2/(2\eta^2)\right)$. Since $f(x) - U_t(x)$ conditioned on $\mathcal{D}_{t-1}$ follows $\mathcal{N}(-\beta_t^{1/2}\sigma_{t-1}(x), \sigma_{t-1}^2(x))$, we have

$$\mathbb{E}[\mathbb{1}[f(x) > U_t(x)] \cdot (f(x) - U_t(x)) \mid \mathcal{D}_{t-1}] \leq \frac{\sigma_{t-1}(x)}{\sqrt{2\pi}} \exp\left(-\frac{\beta_t}{2}\right) \leq \frac{1}{\sqrt{2\pi}|[\mathcal{X}]_t|t^2}, \tag{5.11}$$

where we used $\sigma_t(x) \leq 1$ and $\beta_t = 4(d+1)\log(t) + 2d\log(da'b'\sqrt{\pi}) = 2\log(t^2|[\mathcal{X}]_t|)$. Then,

$$\begin{aligned}
A_2 &= \sum_{t=1}^{T} \mathbb{E}\left[\mathbb{E}[f([X^{I^*}]_t, [X^{\bar{I}^*}]_t) - U_t([X^{I^*}]_t, [X^{\bar{I}^*}]_t) \mid \mathcal{D}_{t-1}] \right] \\
&\leq \sum_{t=1}^{T} \mathbb{E}[\mathbb{E}[\mathbb{1}[f([X^{I^*}]_t, [X^{\bar{I}^*}]_t) - U_t([X^{I^*}]_t, [X^{\bar{I}^*}]_t) \geq 0] \\
&\qquad \cdot (f([X^{I^*}]_t, [X^{\bar{I}^*}]_t) - U_t([X^{I^*}]_t, [X^{\bar{I}^*}]_t)) \mid \mathcal{D}_{t-1}]] \\
&\leq \sum_{t=1}^{T} \mathbb{E}[\mathbb{E}[\sum_{x \in [\mathcal{X}]_t} \mathbb{1}[[f(x) - U_t(x) \geq 0] \cdot (f(x) - U_t(x)) \mid \mathcal{D}_{t-1}]]
\end{aligned}$$

$$\leq \sum_{t=1}^{T} \sum_{x \in [\mathcal{X}]_t} \frac{1}{\sqrt{2\pi}|[\mathcal{X}]_t|t^2}$$
$$\leq \frac{\sqrt{2\pi^3}}{12}. \tag{5.12}$$

Since X^{I^*} and X^{I_t} are identically distributed from the same distribution conditioned on $\mathcal{D}_{t-1}$ and U_t is deterministic, we have

$$A_3 = \sum_{t=1}^{T} \mathbb{E}\Big[\mathbb{E}\Big[U_t([X^{I^*}]_t, [X^{\bar{I}^*}]_t) - U_t([X^{I_t}]_t, [X^{\bar{I}_t}]) \;\Big|\; \mathcal{D}_{t-1}\Big]\Big] = 0. \tag{5.13}$$

Finally, we will bound A_4. Since f and $(X^{I_t}, X^{\bar{I}_t})$ conditioned on $\mathcal{D}_{t-1}$ are independent, we have

$$\begin{aligned}
A_4 &= \sum_{t=1}^{T} \mathbb{E}\Big[\mathbb{E}\Big[U_t([X^{I_t}]_t, [X^{\bar{I}_t}]_t) - f([X^{I_t}]_t, [X^{\bar{I}_t}]_t) \;\Big|\; \mathcal{D}_{t-1}, X^{I_t}, X^{\bar{I}_t}\Big]\Big] \\
&= \sum_{t=1}^{T} \beta_t^{1/2} \mathbb{E}\Big[\sigma_{t-1}([X^{I_t}]_t, [X^{\bar{I}_t}]_t)\Big] \\
&\leq \beta_T^{1/2} \mathbb{E}\Big[\sum_{t=1}^{T} \sigma_{t-1}([X^{I_t}]_t, [X^{\bar{I}_t}]_t)\Big] \\
&\leq \beta_T^{1/2} \mathbb{E}\left[\left(T \sum_{t=1}^{T} \sigma_{t-1}^2([X^{I_t}]_t, [X^{\bar{I}_t}]_t)\right)^{1/2}\right] \\
&\leq \beta_T^{1/2} \mathbb{E}\left[\left(T \frac{2\gamma_T}{\log(1+\sigma^{-2})}\right)^{1/2}\right] \\
&= \sqrt{\frac{2T\beta_T\gamma_T}{\log(1+\sigma^{-2})}},
\end{aligned} \tag{5.14}$$

where we used the monotonic increase of β_t in the first inequality, the Cauchy-Schwarz inequality in the second inequality, and Inequality (5.8) in the third inequality.

By combining Equation (5.9) and Inequalities (5.10), (5.12), (5.13), and (5.14), we complete the proof. □

5.4.2 Proof of Theorem 3

In this section, we give the proof of Theorem 3, where the input distribution is unknown and assumed to be independent of each other variables.

We define an UCB sequence $U_t(\cdot) = \mu_{t-1}(\cdot) + \beta_t^{1/2}\sigma_{t-1}(\cdot)$ in iteration t, where $\beta_t = 4(d+1)\log(t) + 2d\log(da'b'\sqrt{\pi})$. We further define

$$\bar{\bar{g}}_t\left(x^I, x^{\bar{I}}\right) = g_t\left(x^I, x^{\bar{I}}\right) + 2\sum_{i\in\bar{I}} \frac{\alpha_t}{\sqrt{|\mathcal{S}_{t-1}^{(i)}|}}$$
$$X^{J_t} = \underset{x^J\in\mathcal{X}^J, J\in\mathcal{I}}{\operatorname{argmax}} \mathbb{E}_{X^{\bar{J}}\sim F^{\bar{J}}}\left[g_t\left(x^J, X^{\bar{J}}\right)\right].$$

Then, we decompose the regret at iteration t into

$$\begin{aligned}
&\mathbb{E}\left[f\left(X^{I^*}, X^{\bar{I}^*}\right)\right] - \mathbb{E}\left[f\left(X^{I_t}, X^{\bar{I}_t}\right)\right] \\
&= \underbrace{\mathbb{E}\left[f\left(X^{I^*}, X^{\bar{I}^*}\right)\right] - \mathbb{E}\left[\mathbb{E}_{X^{\bar{J}_t}\sim F^{\bar{J}_t}}\left[g_t\left(X^{J_t}, X^{\bar{J}_t}\right)\right]\right]}_{B_{1,t}} \\
&\quad + \underbrace{\mathbb{E}\left[\mathbb{E}_{X^{\bar{J}_t}\sim F^{\bar{J}_t}}\left[g_t\left(X^{J_t}, \bar{J}_t\right)\right]\right] - \mathbb{E}\left[\mathbb{E}_{X^{\bar{J}_t}\sim \hat{F}^{\bar{J}_t}}\left[\bar{g}_t\left(X^{J_t}, X^{\bar{J}_t}\right)\right]\right]}_{B_{2,t}} \\
&\quad + \underbrace{\mathbb{E}\left[\mathbb{E}_{X^{\bar{J}_t}\sim \hat{F}^{\bar{J}_t}}\left[\bar{g}_t\left(X^{J_t}, X^{\bar{J}_t}\right)\right] - \mathbb{E}_{X^{\bar{I}_t}\sim \hat{F}^{\bar{I}_t}}\left[\bar{g}_t\left(X^{I_t}, X^{\bar{I}_t}\right)\right]\right]}_{B_{3,t}} \\
&\quad + \underbrace{\mathbb{E}\left[\mathbb{E}_{X^{\bar{I}_t}\sim \hat{F}^{\bar{I}_t}}\left[\bar{g}_t\left(X^{I_t}, X^{\bar{I}_t}\right)\right]\right] - \mathbb{E}\left[\mathbb{E}_{X^{\bar{I}_t}\sim F^{\bar{I}_t}}\left[\bar{\bar{g}}_t\left(X^{I_t}, X^{\bar{I}_t}\right)\right]\right]}_{B_{4,t}} \\
&\quad + \underbrace{\mathbb{E}\left[\mathbb{E}_{X^{\bar{I}_t}\sim F^{\bar{I}_t}}\left[\bar{\bar{g}}_t\left(X^{I_t}, X^{\bar{I}_t}\right)\right]\right] - \mathbb{E}\left[\mathbb{E}_{X^{\bar{I}_t}\sim F^{\bar{I}_t}}\left[g_t\left(X^{I_t}, X^{\bar{I}_t}\right)\right]\right]}_{B_{5,t}} \\
&\quad + \underbrace{\mathbb{E}\left[\mathbb{E}_{X^{\bar{I}_t}\sim F^{\bar{I}_t}}\left[g_t\left(X^{I_t}, X^{\bar{I}_t}\right)\right]\right] - \mathbb{E}\left[\mathbb{E}_{X^{\bar{I}_t}\sim F^{\bar{I}_t}}\left[U_t\left(X^{I_t}, X^{\bar{I}_t}\right)\right]\right]}_{B_{6,t}} \\
&\quad + \underbrace{\mathbb{E}\left[\mathbb{E}_{X^{\bar{I}_t}\sim F^{\bar{I}_t}}\left[U_t\left(X^{I_t}, X^{\bar{I}_t}\right)\right]\right] - \mathbb{E}\left[f\left(X^{I_t}, X^{\bar{I}_t}\right)\right]}_{B_{7,t}}. \qquad (5.15)
\end{aligned}$$

First, since f and g_t follow the same distribution given $\mathcal{D}_{t-1}$, we have

$$\begin{aligned}
\mathbb{E}\left[f\left(X^{I^*}, X^{\bar{I}^*}\right)\right] &= \mathbb{E}\left[\max_{x^I} \mathbb{E}_{X^{\bar{I}} \sim F^{\bar{I}}}\left[f\left(x^I, X^{\bar{I}}\right)\right]\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[\max_{x^I} \mathbb{E}_{X^{\bar{I}} \sim F^{\bar{I}}}\left[f\left(x^I, X^{\bar{I}}\right)\right] | \mathcal{D}_{t-1}\right]\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[\max_{x^I} \mathbb{E}_{X^{\bar{I}} \sim F^{\bar{I}}}\left[g_t\left(x^I, X^{\bar{I}}\right)\right] | \mathcal{D}_{t-1}\right]\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[\mathbb{E}_{X^{\bar{J}_t} \sim F^{\bar{J}_t}}\left[g_t\left(X^{J_t}, X^{\bar{J}_t}\right)\right] | \mathcal{D}_{t-1}\right]\right] \\
&= \mathbb{E}\left[\mathbb{E}_{X^{\bar{J}_t} \sim F^{\bar{J}_t}}\left[g_t\left(X^{J_t}, X^{\bar{J}_t}\right)\right]\right],
\end{aligned}$$

for which we have

$$B_{1,t} = 0. \tag{5.16}$$

Next, we will upper-bound $B_{2,t}$ and $B_{4,t}$. We decompose $B_{2,t}$ into

$$\begin{aligned}
B_{2,t} = &\mathbb{E}\left[\mathbb{1}\left[\|g_t'\|_\infty \le \zeta_t\right]\left(\mathbb{E}_{X^{\bar{J}_t} \sim F^{\bar{J}_t}}\left[g_t\left(X^{J_t}, X^{\bar{J}_t}\right)\right] - \mathbb{E}_{X^{\bar{J}_t} \sim \hat{F}^{\bar{J}_t}}\left[g_t\left(X^{J_t}, X^{\bar{J}_t}\right)\right]\right)\right] \\
&- \mathbb{E}\left[\sum_{i \in \bar{J}_t} \frac{\alpha_t}{\sqrt{|\mathcal{S}_t^{(i)}|}}\right] \\
&+ \mathbb{E}\left[\mathbb{1}\left[\|g_t'\|_\infty > \zeta_t\right]\left(\mathbb{E}_{X^{\bar{J}_t} \sim F^{\bar{J}_t}}\left[g_t\left(X^{J_t}, X^{\bar{J}_t}\right)\right] - \mathbb{E}_{X^{\bar{J}_t} \sim \hat{F}^{\bar{J}_t}}\left[g_t\left(X^{J_t}, X^{\bar{J}_t}\right)\right]\right)\right] \\
\le &\mathbb{E}\left[\mathbb{1}\left[\|g_t'\|_\infty \le \zeta_t\right]\left(\mathbb{E}_{X^{\bar{J}_t} \sim F^{\bar{J}_t}}\left[g_t\left(X^{J_t}, X^{\bar{J}_t}\right)\right] - \mathbb{E}_{X^{\bar{J}_t} \sim \hat{F}^{\bar{J}_t}}\left[g_t\left(X^{J_t}, X^{\bar{J}_t}\right)\right]\right)\right] \\
&- \mathbb{E}\left[\sum_{i \in \bar{J}_t} \frac{\alpha_t}{\sqrt{|\mathcal{S}_t^{(i)}|}}\right] + 2\mathbb{E}\left[\mathbb{1}\left[\|g_t'\|_\infty > \zeta_t\right] \|g_t\|_\infty\right] \\
\le &\mathbb{E}\left[\mathbb{1}\left[\|g_t'\|_\infty \le \zeta_t\right]\left(\mathbb{E}_{X^{\bar{J}_t} \sim F^{\bar{J}_t}}\left[g_t\left(X^{J_t}, X^{\bar{J}_t}\right)\right] - \mathbb{E}_{X^{\bar{J}_t} \sim \hat{F}^{\bar{J}_t}}\left[g_t\left(X^{J_t}, X^{\bar{J}_t}\right)\right]\right)\right]
\end{aligned}$$

$$
\begin{aligned}
&- \mathbb{E}\left[\sum_{i\in\bar{J}_t} \frac{\alpha_t}{\sqrt{|\mathcal{S}_t^{(i)}|}}\right] \\
&+ 2\eta_t \mathbb{E}\left[\mathbb{P}\left[\|g_t'\|_\infty > \zeta_t | \mathcal{D}_{t-1}\right]\right] + 2\mathbb{E}\left[\mathbb{E}\left[\mathbb{1}\left[\|g_t\|_\infty > \eta_t\right]\|g_t\|_\infty | \mathcal{D}_{t-1}\right]\right] \\
&= \underbrace{\mathbb{E}\left[\mathbb{1}\left[\|g_t'\|_\infty \le \zeta_t\right]\left(\mathbb{E}_{X^{\bar{J}_t}\sim F^{\bar{J}_t}}\left[g_t\left(X^{J_t}, X^{\bar{J}_t}\right)\right] - \mathbb{E}_{X^{\bar{J}_t}\sim \hat{F}^{\bar{J}_t}}\left[g_t\left(X^{J_t}, X^{\bar{J}_t}\right)\right]\right)\right]}_{B_{2\text{-}1,t}} \\
&\underbrace{-\mathbb{E}\left[\sum_{i\in\bar{J}_t} \frac{\alpha_t}{\sqrt{|\mathcal{S}_{t-1}^{(i)}|}}\right]}_{B_{2\text{-}2,t}} + \underbrace{2\eta_t \mathbb{P}\left[\|f'\|_\infty > \zeta_t\right]}_{B_{2\text{-}3,t}} + \underbrace{2\mathbb{E}\left[\mathbb{1}\left[\|f\|_\infty > \eta_t\right]\|f\|_\infty\right]}_{B_{2\text{-}4,t}}, \qquad (5.17)
\end{aligned}
$$

where $\|g_t'\|_\infty := \sup_{x\in\mathcal{X}, j\in[d]} |\partial g_t(x)/\partial x^{(j)}|$, $\zeta_t := b'\sqrt{2\log t}$, $\eta_t := b\sqrt{2\log(t+1)}$, and $b', b > 0$ are the constants in Assumptions (5.7) and (5.4). Here, the last equality is because f and g_t' are independently distributed from the same distribution given $\mathcal{D}_{t-1}$.

Using the independence assumption of the input random variables and integration by parts, we have

$$
\begin{aligned}
&\mathbb{E}_{X^{\bar{J}_t}\sim F^{\bar{J}_t}}\left[g_t\left(X^{J_t}, X^{\bar{J}_t}\right)\right] - \mathbb{E}_{X^{\bar{J}_t}\sim \hat{F}^{\bar{J}_t}}\left[g_t\left(X^{J_t}, X^{\bar{J}_t}\right)\right] \\
&= \sum_{i=1}^{|\bar{J}_t|} \mathbb{E}_{X^{\bar{J}_t\setminus\bar{J}_{t,i}}}\left[\int_0^1 g_t\left(x^{\bar{J}_{t,i}}, X^{\bar{J}_t\setminus\bar{J}_{t,i}}\right)\left(dF^{\bar{J}_{t,i}}(x^{\bar{J}_{t,i}}) - d\hat{F}^{\bar{J}_{t,i}}(x^{\bar{J}_{t,i}})\right)\right] \\
&= \sum_{i=1}^{|\bar{J}_t|} \mathbb{E}_{X^{\bar{J}_t\setminus\bar{J}_{t,i}}}\left[-\int_0^1 \frac{\partial g_t}{\partial x^{\bar{J}_{t,i}}}\left(x^{\bar{J}_{t,i}}, X^{\bar{J}_t\setminus\bar{J}_{t,i}}\right)\left(F^{\bar{J}_{t,i}}(x^{\bar{J}_{t,i}}) - \hat{F}^{\bar{J}_{t,i}}(x^{\bar{J}_{t,i}})\right) dx^{\bar{J}_{t,i}}\right] \\
&\le \|g_t'\|_\infty \sum_{i\in\bar{J}_t} e^{(i)}_{|\mathcal{S}_{t-1}^{(i)}|}, \qquad (5.18)
\end{aligned}
$$

where

$$
e^{(i)}_{|\mathcal{S}_{t-1}^{(i)}|} := \sup_{x^{(i)}\in\mathcal{X}^{(i)}} \left|F^{(i)}(x^{(i)}) - \hat{F}^{(i)}_{|\mathcal{S}_{t-1}^{(i)}|}\right|
$$

and $\mathbb{E}_{X^{\bar{J}_t \setminus \bar{J}_{t,i}}}$ denotes an expectation operator for abbreviation defined as

$$\mathbb{E}_{X^{\bar{J}_t \setminus \bar{J}_{t,i}}} := \mathbb{E}_{X^{\bar{J}_{t,1}} \sim F^{\bar{J}_{t,1}}} \cdots \mathbb{E}_{X^{\bar{J}_{t,i-1}} \sim F^{\bar{J}_{t,i-1}}} \mathbb{E}_{X^{\bar{J}_{t,i+1}} \sim \hat{F}^{\bar{J}_{t,i+1}}} \cdots \mathbb{E}_{X^{\bar{J}_{t,|\bar{I}|}} \sim \hat{F}^{\bar{J}_{t,|\bar{J}_t|}}}.$$

Using Inequality (5.18), we have

$$\begin{aligned} B_{2\text{-}1,t} + B_{2\text{-}2,t} &\leq \mathbb{E}\left[\mathbb{1}\left[\|g_t'\|_\infty \leq \zeta_t\right] \|g_t'\|_\infty \sum_{i \in \bar{J}_t} e^{(i)}_{|\mathcal{S}^{(i)}_{t-1}|}\right] - \mathbb{E}\left[\sum_{i \in \bar{J}_t} \frac{\alpha_t}{\sqrt{|\mathcal{S}^{(i)}_{t-1}|}}\right] \\ &\leq \zeta_t \sum_{i \in \bar{J}_t} \mathbb{E}\left[e^{(i)}_{|\mathcal{S}^{(i)}_{t-1}|} - \sqrt{\frac{2\log t}{|\mathcal{S}^{(i)}_{t-1}|}}\right] \end{aligned} \tag{5.19}$$

Since samples $x^i \in \mathcal{S}^i_{t-1}$ for $i \in \bar{J}_t$ are independently distributed given $|\mathcal{S}^i_{t-1}|$ and F^i is the continuous cumulative distribution function for the assumption, by applying the Dvoretzky–Kiefer–Wolfowitz inequality [16, 50] conditioned on $n = |\mathcal{S}^i_{t-1}|$, for $\epsilon > 0$, we have

$$P^{(i)}_n(\epsilon) := \mathbb{P}\left(e^{(i)}_{|\mathcal{S}^{(i)}_{t-1}|} > \epsilon \;\middle|\; |\mathcal{S}^i_{t-1}| = n\right) \leq 2\exp\left(-2n\epsilon^2\right).$$

Define $\epsilon_{t,n} := \epsilon + \sqrt{(2\log t)/n}$ for $\epsilon > 0$, and we have

$$\begin{aligned} &\mathbb{E}\left[e^{(i)}_{|\mathcal{S}^{(i)}_{t-1}|} - \sqrt{\frac{2\log t}{\left|\mathcal{S}^{(i)}_{t-1}\right|}}\right] \\ &= \mathbb{E}\left[\mathbb{1}\left[e^{(i)}_{|\mathcal{S}^{(i)}_{t-1}|} - \sqrt{\frac{2\log t}{\left|\mathcal{S}^{(i)}_{t}\right|}} \leq \epsilon\right]\left(e^{(i)}_{|\mathcal{S}^{(i)}_{t-1}|} - \sqrt{\frac{2\log t}{\left|\mathcal{S}^{(i)}_{t}\right|}}\right)\right] \\ &\quad + \mathbb{E}\left[\mathbb{1}\left[e^{(i)}_{|\mathcal{S}^{(i)}_{t-1}|} - \sqrt{\frac{2\log t}{\left|\mathcal{S}^{(i)}_{t}\right|}} > \epsilon\right]\left(e^{(i)}_{|\mathcal{S}^{(i)}_{t-1}|} - \sqrt{\frac{2\log t}{\left|\mathcal{S}^{(i)}_{t}\right|}}\right)\right] \\ &\leq \epsilon + \sum_{n=1}^{T} \mathbb{E}\left[\mathbb{1}\left[e^{(i)}_{|\mathcal{S}^{(i)}_{t-1}|} - \sqrt{\frac{2\log t}{n}} > \epsilon, \left|\mathcal{S}^{(i)}_{t}\right| = n\right]\left(e^{(i)}_{|\mathcal{S}^{(i)}_{t-1}|} - \sqrt{\frac{2\log t}{n}}\right)\right] \end{aligned}$$

$$
\begin{aligned}
&\leq \epsilon + \sum_{n=1}^{T} \int_{\epsilon_{t,n}}^{\infty} \left(x - \sqrt{\frac{2\log t}{n}} \right) \mathrm{d}\left(-P_n^{(i)}(x) \right) \\
&\leq \epsilon + \sum_{n=1}^{T} \left(- \left[\left(x - \sqrt{\frac{2\log t}{n}} \right) P_n^{(i)}(x) \right]_{\epsilon_{t,n}}^{\infty} + \int_{\epsilon_{t,n}}^{\infty} P_n^{(i)}(x) \mathrm{d}x \right) \\
&\leq \epsilon + \sum_{n=1}^{T} \left(2\epsilon \mathrm{e}^{-2n\epsilon_{t,n}^2} + \int_{\epsilon_{t,n}}^{\infty} \frac{1}{nx} \cdot 2nx \mathrm{e}^{-2nx^2} \mathrm{d}x \right) \\
&= \epsilon + \sum_{n=1}^{T} \left(2\epsilon \mathrm{e}^{-n\epsilon_{t,n}^2} + \left[-\frac{1}{nx} \cdot \mathrm{e}^{-nx^2} \right]_{\epsilon_{t,n}}^{\infty} - \int_{\epsilon_{t,n}}^{\infty} \frac{1}{nx^2} \cdot \mathrm{e}^{-nx^2} \mathrm{d}x \right) \\
&\leq \epsilon + 2\epsilon \sum_{n=1}^{T} \mathrm{e}^{-n\epsilon_{t,n}^2} + \sum_{n=1}^{T} \frac{1}{n\epsilon_{t,n}} \mathrm{e}^{-n\epsilon_{t,n}^2} \\
&\leq \epsilon + 2\epsilon \sum_{n=1}^{T} \mathrm{e}^{-n\epsilon^2 - 2\log t} + \sum_{n=1}^{T} \frac{1}{n\epsilon} \mathrm{e}^{-n\epsilon^2 - 2\log t} \\
&= \epsilon + \frac{2\epsilon}{t^2} \sum_{n=1}^{\infty} \mathrm{e}^{-n\epsilon^2} + \frac{1}{\epsilon t^2} \sum_{n=1}^{T} \frac{1}{n} \mathrm{e}^{-n\epsilon^2} \\
&\leq \epsilon + \frac{2\epsilon}{t^2} \cdot \frac{1}{e^{\epsilon^2} - 1} - \frac{1}{\epsilon t^2} \log(1 - \mathrm{e}^{-\epsilon^2}) && (5.20) \\
&\leq \epsilon + \frac{2\epsilon}{t^2} \cdot \frac{1}{\epsilon^2} + \frac{1}{\epsilon t^2} \left(\epsilon^2 + \log \left(\frac{1}{\mathrm{e}^{\epsilon^2} - 1} \right) \right) && (5.21) \\
&\leq \epsilon + \frac{2}{\epsilon t^2} + \frac{\epsilon}{t^2} + \frac{1}{\epsilon t^2} \log \frac{1}{\epsilon^2} && (5.22) \\
&\leq \epsilon + \frac{2}{\epsilon t^2} + \frac{\epsilon}{t^2} + \frac{1}{\epsilon t^2} \log \left(\frac{1}{\epsilon^2} \right), && (5.23)
\end{aligned}
$$

where we used $\sum_{n=1}^{\infty} x^n/n = -\log(1-x)$ in Equality (5.20) and $e^x \geq x + 1$ in Inequalities (5.22) and (5.21). By taking the summation over T and combining Inequalities (5.19) and (5.23), we have

$$
\begin{aligned}
\sum_{t=1}^{T} (B_{2\text{-}1,t} + B_{2\text{-}2,t}) &\leq \sum_{t=1}^{T} \zeta_t \sum_{i \in \bar{J}_t} \left(\epsilon + \frac{2}{\epsilon t^2} + \frac{\epsilon}{t^2} + \frac{1}{\epsilon t^2} \log \left(\frac{1}{\epsilon^2} \right) \right) \\
&\leq b' m \sqrt{2\log T} \left(\epsilon T + \frac{\pi^2}{3\epsilon} + \frac{\pi^2}{6} \epsilon + \frac{\pi^2}{6\epsilon} \log \left(\frac{1}{\epsilon^2} \right) \right), && (5.24)
\end{aligned}
$$

where $m = \max_{I \in \mathcal{I}} |\bar{I}|$.

$B_{2\text{-}3,t}$ after taking summation over T is upper-bounded because of Assumption (5.4).

$$\begin{aligned}\sum_{t=1}^{T} B_{2\text{-}3,t} &\leq \sum_{t=1}^{T} 2\eta_t a' e^{-(\zeta_t/b')^2} \\ &\leq 2a'b\sqrt{2\log(T+1)}\sum_{t=1}^{T}\frac{1}{t^2} \\ &\leq \frac{\pi^2 a' b\sqrt{2\log(T+1)}}{3}, \end{aligned} \tag{5.25}$$

where we used monotonic increase of B in the second inequality.

The term $B_{2\text{-}4,t}$ is upper-bounded as

$$\begin{aligned} B_{2\text{-}4,t} &= 2\mathbb{E}\left[\int_0^\infty \mathbb{1}\left[\mathbb{1}\left[\|f\|_\infty > \eta_t\right]\cdot\|f\|_\infty > z\right] dz\right] \\ &= 2\mathbb{E}\left[\int_0^\infty \mathbb{1}\left[\|f\|_\infty > \eta_t\right]\cdot\mathbb{1}\left[\|f\|_\infty > z\right] dz\right] \\ &= 2\mathbb{E}\left[\int_0^{\eta_t} \mathbb{1}\left[\|f\|_\infty > \eta_t\right] dz + \int_{\eta_t}^\infty \mathbb{1}\left[\|f\|_\infty > z\right] dz\right] \\ &= 2\eta_t\mathbb{P}\left[\|f\|_\infty > \eta_t\right] + 2\int_{\eta_t}^\infty \mathbb{P}\left(\|f\|_\infty > z\right) dz \\ &\leq 2\eta_t a e^{-(\eta_t/b)^2} + 2\int_{\eta_t}^\infty a e^{-(z/b)^2} dz \qquad (5.26) \\ &\leq 2\eta_t a e^{-(\eta_t/b)^2} + 2a\int_{\eta_t}^\infty \frac{z}{\eta_t} e^{-(z/b)^2} dz \\ &= 2\eta_t a e^{-(\eta_t/b)^2} + \frac{ab^2}{\eta_t} e^{-(\eta_t/b)^2} \\ &= \left(2\eta_t + \frac{b^2}{\eta_t}\right) a e^{-(\eta_t/b)^2}, \qquad (5.27) \end{aligned}$$

where we used Assumptions (5.7) and (5.4) in Inequality (5.26). Therefore, by taking summation over T, we have

$$\sum_{t=1}^{T} B_{2\text{-}4,t} \leq \sum_{t=1}^{T}\left(2\eta_t + \frac{b^2}{\eta_t}\right) a e^{-(\eta_t/b)^2}$$

$$\leq \left(2\sqrt{2\log(T+1)} + \frac{1}{\sqrt{2\log 2}}\right) ab \sum_{t=1}^{T} \frac{1}{t^2+1}$$
$$= \frac{\pi^2 ab\sqrt{2\log(T+1)}}{3} + \frac{\pi^2 ab}{6\sqrt{2\log 2}}, \tag{5.28}$$

where we used Inequality (5.27) in the first inequality and the monotonic increase of η_t in the second inequality.

By combining Inequalities (5.17), (5.24), (5.25) and (5.28), we obtain

$$\sum_{t=1}^{T} B_{2,t} \leq \frac{\pi^2 ab}{6\sqrt{2\log 2}} + \frac{\pi^2 b(a+a')\sqrt{2\log(T+1)}}{3}$$
$$+ b'm\sqrt{2\log T}\left(\epsilon T + \frac{\pi^2}{3\epsilon} + \frac{\pi^2}{6}\epsilon + \frac{\pi^2}{6\epsilon}\log\left(\frac{1}{\epsilon^2}\right)\right) \tag{5.29}$$

Applying the same argument to $\sum_{t=1}^{T} B_{4,t}$, we obtain the same bound.

Due to the definition of the acquisition function given $\mathcal{D}_{t-1}$, we have

$$B_{3,t} \leq 0. \tag{5.30}$$

Next, we upper-bound $B_{5,t}$. We have

$$\sum_{t=1}^{T} B_{5,t} \leq 2\alpha_T \mathbb{E}\left[\sum_{t=1}^{T}\sum_{i\in\bar{I}_t} \frac{1}{\sqrt{|\mathcal{S}_{t-1}^{(i)}|}}\right]$$
$$= 2\alpha_T \mathbb{E}\left[\sum_{n=1}^{T}\sum_{i=1}^{d}\sum_{t=1}^{T} \frac{\mathbb{1}\left[i\in\bar{I}_t, |\mathcal{S}_{t-1}^{(i)}| = n\right]}{\sqrt{n}}\right]$$
$$= 2\alpha_T \mathbb{E}\left[\sum_{n=1}^{T}\sum_{i=1}^{d} \frac{\mathbb{1}\left[\bigcup_{t=1}^{T}\left\{i\in\bar{I}_t, |\mathcal{S}_{t-1}^{(i)}| = n\right\}\right]}{\sqrt{n}}\right] \tag{5.31}$$
$$\leq 2\alpha_T \mathbb{E}\left[\sum_{n=1}^{\lceil mT/d\rceil}\sum_{i=1}^{d} \frac{1}{\sqrt{n}}\right] \tag{5.32}$$

$$\leq 2d\alpha_T \int_0^{mT/d+1} \frac{1}{\sqrt{x}} dx$$
$$\leq 8b'\sqrt{d(mT+d)}\log T, \tag{5.33}$$

where we used monotonic increase of α_t in the first inequality. Equality (5.31) is because events $\{i \in \bar{I}_t, |\mathcal{S}_{t-1}^{(i)}| = n\}$ for $n = 1, \ldots, T$ happen once at most over $t = 1, \ldots, T$. Inequality (5.32) is because $\sum_{n=1}^{T}\sum_{i=1}^{d} \mathbb{1}\left[\bigcup_{t=1}^{T}\left\{i \in \bar{I}_t, |\mathcal{S}_{t-1}^{(i)}| = n\right\}\right] \leq mT$ and $1/\sqrt{n}$ is monotonically decreasing in n.

To upper-bound the term $B_{6,t}$, we decomposed it into three terms as

$$\begin{aligned} B_{6,t} = &\underbrace{\mathbb{E}\left[\mathbb{E}\left[\mathbb{E}_{X^{\bar{I}_t}\sim F^{\bar{I}_t}}\left[g_t\left(X^{I_t}, X^{\bar{I}_t}\right) - g_t\left([X^{I_t}]_t, [X^{\bar{I}_t}]_t\right)\right]\middle|\mathcal{D}_{t-1}\right]\right]}_{B_{6\text{-}1,t}} \\ &+ \underbrace{\mathbb{E}\left[\mathbb{E}\left[\mathbb{E}_{X^{\bar{I}_t}\sim F^{\bar{I}_t}}\left[g_t\left([X^{I_t}]_t, [X^{\bar{I}_t}]_t\right) - U_t\left([X^{I_t}]_t, [X^{\bar{I}_t}]_t\right)\right]\middle|\mathcal{D}_{t-1}\right]\right]}_{B_{6\text{-}2,t}} \\ &+ \underbrace{\mathbb{E}\left[\mathbb{E}\left[\mathbb{E}_{X^{\bar{I}_t}\sim F^{\bar{I}_t}}\left[\left(U_t\left([X^{I_t}]_t, [X^{\bar{I}_t}]_t\right) - U_t(X^{I_t}, X^{\bar{I}_t})\right)\right]\middle|\mathcal{D}_{t-1}\right]\right]}_{B_{6\text{-}3,t}}. \end{aligned} \tag{5.34}$$

Because f and g_t' are independently distributed from the same distribution given $\mathcal{D}_{t-1}$, we upper-bound the term $B_{6\text{-}1,t}$ after summing over T as

$$\begin{aligned} \sum_{t=1}^{T} B_{6\text{-}1,t} &\leq \sum_{t=1}^{T} \frac{d}{\tau_t}\mathbb{E}\left[\mathbb{E}\left[\sup_{x\in\mathcal{X}, j\in[d]}\left|\frac{\partial g_t(x)}{\partial x^{(j)}}\right| \mid \mathcal{D}_{t-1}\right]\right] \\ &= \sum_{t=1}^{T} \frac{d}{\tau_t}\mathbb{E}\left[\mathbb{E}\left[\sup_{x\in\mathcal{X}, j\in[d]}\left|\frac{\partial f(x)}{\partial x^{(j)}}\right| \mid \mathcal{D}_{t-1}\right]\right] \\ &= \sum_{t=1}^{T} \frac{d}{\tau_t}\mathbb{E}\left[\sup_{x\in\mathcal{X}, j\in[d]}\left|\frac{\partial f(x)}{\partial x^{(j)}}\right|\right] \\ &\leq \frac{\pi^2}{12}, \end{aligned} \tag{5.35}$$

where we applied the same argument as Inequality (5.10) in the last inequality.

Since $g_t(x) - U_t(x)$ conditioned on $\mathcal{D}_{t-1}$ follows $\mathcal{N}(-\beta_t^{1/2}\sigma_{t-1}(x), \sigma_{t-1}^2(x))$, by applying

the same argument as Inequality (5.12), we have

$$\begin{aligned}
\sum_{t=1}^{T} B_{6\text{-}2,t} &\leq \sum_{t=1}^{T} \mathbb{E}[\mathbb{E}[\mathbb{1}[g_t([X^{I_t}]_t, [X^{\bar{I}_t}]_t) - U_t([X^{I_t}]_t, [X^{\bar{I}_t}]_t) \geq 0] \\
&\qquad\qquad \cdot (g_t([X^{I_t}]_t, [X^{\bar{I}_t}]_t) - U_t([X^{I_t}]_t, [X^{\bar{I}_t}]_t)) \mid \mathcal{D}_{t-1}]] \\
&= \sum_{t=1}^{T} \mathbb{E}[\mathbb{E}[\sum_{x \in [\mathcal{X}]_t} \mathbb{1}[g_t(x) - U_t(x) \geq 0] \cdot (g_t(x) - U_t(x)) \mid \mathcal{D}_{t-1}]] \\
&\leq \sum_{t=1}^{T} \mathbb{E}[\mathbb{E}[\sum_{x \in [\mathcal{X}]_t} \mathbb{1}[f(x) - U_t(x) \geq 0] \cdot (f(x) - U_t(x)) \mid \mathcal{D}_{t-1}]] \\
&\leq \sum_{t=1}^{T} \sum_{x \in [\mathcal{X}]_t} \frac{1}{\sqrt{2\pi}|[\mathcal{X}]_t|t^2} \\
&\leq \frac{\sqrt{2\pi^3}}{12}.
\end{aligned} \tag{5.36}$$

Since f and $(X^{I_t}, X^{\bar{I}_t})$ are independent given $\mathcal{D}_{t-1}$, we upper-bound $B_{6\text{-}3,t}$ as

$$\begin{aligned}
\sum_{t=1}^{T} B_{6\text{-}3,t} &\leq \sum_{t=1}^{T} \mathbb{E}\left[\mathbb{E}\left[\left|f([X^{I_t}]_t, [X^{\bar{I}}]_t) - f(X^{I_t}, X^{\bar{I}_t})\right| \mid X^{I_t}, X^{\bar{I}_t}, \mathcal{D}_{t-1}\right]\right] \\
&\quad + \sum_{t=1}^{T} \mathbb{E}\left[\beta_t^{1/2}(\sigma_{t-1}([X^{I_t}]_t, [X^{\bar{I}_t}]_t) - \sigma_{t-1}(X^{I_t}, X^{\bar{I}_t}))\right] \\
&\leq \sum_{t=1}^{T} \mathbb{E}\left[\frac{d}{\tau_t} \sup_{x \in \mathcal{X}, j \in [d]} \left|\frac{\partial f(x)}{\partial x^{(j)}}\right|\right] + \sum_{t=1}^{T} \beta_t^{1/2} \mathbb{E}\left[\sigma_{t-1}([X^{I_t}]_t, [X^{\bar{I}_t}]_t)\right] \\
&\leq \frac{\pi^2}{12} + \sqrt{\frac{2T\beta_T\gamma_T}{\log(1+\sigma^{-2})}},
\end{aligned} \tag{5.37}$$

where we applied the same argument as Inequalities (5.10) and (5.14) in the last inequality.

By combining Equation (5.34) and Inequalities (5.35), (5.36), and (5.37), we have

$$\sum_{t=1}^{T} B_{6,t} \leq \frac{\pi^2}{6} + \frac{\sqrt{2\pi^3}}{12} + \sqrt{\frac{2T\beta_T\gamma_T}{\log(1+\sigma^{-2})}} \tag{5.38}$$

Since f and $(X^{I_t}, X^{\bar{I}_t})$ given $\mathcal{D}_{t-1}$ are independent, by applying the same argument as (5.14) to $B_{7,t}$ over T, we have

$$\sum_{t=1}^{T} B_{7,t} \leq \sqrt{\frac{2T\beta_T\gamma_T}{\log(1+\sigma^{-2})}}. \tag{5.39}$$

Finally, we combine Equation (5.15) and Inequalities (5.16), (5.29), (5.30), (5.33), (5.38), and (5.39), and we have

$$\begin{aligned}\text{BayesRegret}(T) \leq & \frac{\pi^2}{6} + \frac{\sqrt{2\pi^3}}{12} + \frac{\pi^2 ab}{3\sqrt{2\log 2}} \\ & + \frac{2\pi^2(a+a')'b\sqrt{2\log(T+1)}}{3} + 8b'\sqrt{d(mT+d)}\log T \\ & + 2b'm\sqrt{2\log T}\left(\epsilon T + \frac{\pi^2}{3\epsilon} + \frac{\pi^2}{6}\epsilon + \frac{\pi^2}{6\epsilon}\log\left(\frac{1}{\epsilon^2}\right)\right) + 2\sqrt{\frac{2T\beta_T\gamma_T}{\log(1+\sigma^{-2})}}\end{aligned} \tag{5.40}$$

By setting $\epsilon = \sqrt{\log T/T}$, we have

$$\begin{aligned}\text{BayesRegret}(T) \leq & \frac{\pi^2}{6} + \frac{\sqrt{2\pi^3}}{12} + \frac{\pi^2 ab}{3\sqrt{2\log 2}} + \frac{\sqrt{2}\pi^2 b'm}{3}\frac{\log T}{\sqrt{T}} \\ & + \frac{2\pi^2(a+a')'b\sqrt{2\log(T+1)}}{3} + \frac{2\pi^2 b'm\sqrt{2T}}{3} + \frac{\pi^2 b'm\sqrt{2T}}{3}\log\left(\frac{T}{\log T}\right) \\ & + 2b'm\sqrt{2T\log T} + 8b'\sqrt{d(mT+d)}\log T + 2\sqrt{\frac{2dT\beta_T\gamma_T}{\log(1+\sigma^{-2})}} \\ & \in O\left(\sqrt{dT\log T}(\sqrt{\gamma_T} + \sqrt{m\log T})\right),\end{aligned} \tag{5.41}$$

which completes the proof. □

5.5 Experiments

We demonstrate the effectiveness of the proposed algorithms for the known and unknown input distributions using a set of test functions and real-world datasets.

5.5.1 Comparing Methods

We compare the proposed algorithms to four baselines: DropoutBO, WrapperBOs, RandomBO, and Random.

DropoutBO [45] was originally proposed for the high-dimensional BO, and it sequentially determines a partially specified query by performing optimization over a low-dimensional input domain. In each iteration, DropoutBO randomly selects $I_t \in \mathcal{I}$ and then performs the standard BO over $\mathcal{X}^I$ to determine x^{I_t}.

WrapperBOs performs the standard BO for each $I \in \mathcal{I}$. It applies $|\mathcal{I}|$ GP models $\{\mathcal{GP}^I\}_{I\in\mathcal{I}}$ to a set of datasets $\{\{(x_i^I, y_i)\}_{i=1}^t\}_{I\in\mathcal{I}}$, respectively, where $x_i = (x^{I_i}, X^{\bar{I}_i})$. It determines the next control variables as $x^{I_t} = \arg\max_{I\in\mathcal{I}, x^I\in\mathcal{X}^I} a^I(x^I)$, where $a^I : \mathcal{X}^I \to \mathbb{R}$ is the acquisition function associated with $\mathcal{GP}^I$. WrapperBOs can be considered a wrapper method [26] in the literature of feature selection. This method is computationally inefficient because it treats each $I \in \mathcal{I}$ independently and repeats the fitting of the GP models $|\mathcal{I}|$ (which is possibly an exponential number) times.

RandomBO directly uses the standard BO method with a randomly determined control variable set. First, it determines the next d-dimensional input as in the standard BO, $x_t = \arg\max_{x\in\mathcal{X}} a(x)$. Then, I_t is randomly selected from a uniform distribution over $\mathcal{I}$. Next, it queries for $x_t^{I_t}$.

Random simply determines the next control variables according to a uniform distribution.

We use the Gaussian kernel function for GP models and employ Thompson sampling for the methods.

5.5.2 Approximation of Sample Paths

All methods except RANDOM use a sample path $g_t \sim \mathcal{GP}(\mu_{t-1}, k_{t-1})$. By approximating the GP with a finite-dimensional Bayesian linear model, we can efficiently optimize g_t [27]. Bochner's theorem [61] guarantees that for any continuous shift-invariant kernel $k(x-y)$, its Fourier transform is a non-negative measure $\hat{k}(\omega)/\alpha$, where $\alpha = \int_\omega \hat{k}(\omega) d\omega$ denotes a normalization constant. The principle of random Fourier features [59] is to approximate the kernel function as

$$k(x-y) = \alpha \mathbb{E}[e^{-i\omega^\top (x-y)}] \approx \Phi(x)^\top \Phi(y),$$

where

$$\Phi(x) = \sqrt{\frac{2\alpha}{M}} \begin{pmatrix} \sin(\omega_1^\top x) \\ \cos(\omega_1^\top x) \\ \vdots \\ \sin(\omega_{M/2}^\top x) \\ \cos(\omega_{M/2}^\top x) \end{pmatrix}$$

and $\{\omega_i\}_{i=1}^{M/2} \overset{\text{i.i.d.}}{\sim} \hat{k}(\omega)$. Then, the GP is approximated with an M-dimensional Bayesian linear model. The sample path is approximated as

$$g_t(x) \approx \Phi(x)^\top \theta_t, \quad \theta_t \sim \mathcal{N}(\tilde{\mu}_t, \tilde{\Sigma}_t),$$

where $\tilde{\mu}_t = A^{-1} \Phi(X_{t-1})^\top Y_{t-1}$, $\tilde{\Sigma}_t = \sigma^2 A^{-1}$, $A = \Phi(X_{t-1})^\top \Phi(X_{t-1}) + \sigma^2 I$, $\Phi(X_{t-1}) = (\Phi(x_1), \dots \Phi(x_{t-1}))^\top \in \mathbb{R}^{(t-1)\times M}$, and $Y_{t-1} = (y_1, \dots, y_{t-1})^\top$. The random Fourier features enable us to deal with a large number of samples. In the experiments, we set $M = 1000$.

5.5.3 Experiments Using a Test Function with Fixed Parameters

In these experiments, we study the cases of the known and unknown input distributions where parameters are fixed to the ground-truth ones. We use the Branin-Hoo function [58] (Figure 5.1(left)) over a two dimensional input domain as a black-box target function without observation noise. We model a joint distribution of input variables with a Gaussian distribution $\mathcal{N}(\mu, \Sigma)$ truncated over $[0, 1]^2$, where we set $\mu = (0.5, 0.5)^\top$ and $\Sigma = ((0.01, 0)^\top, (0, 0.05)^\top)$ (Figure 5.1 (right)). We set a family of control variable sets as $\mathcal{I} = \{\{1\}, \{2\}\}$. The Branin-Hoo function has three minimum points (red dots in Figure 5.1 (left)); however, the optimal control variable is different (red line in Figure 5.1 (right)). We fix the hyperparameters of the kernel function to those obtained by the type I I maximum likelihood estimation using thousands of data points as the ground-truth ones. TSPSQ-UNKNOWN for the unknown input distribution has a hyperparameter $c > 0$ in $\alpha_t = c \log t$ in Equation (5.6) to control the exploration-exploitation trade-off in the distribution estimation. To select a practical value for the unknown input distribution case, we conduct a sensitivity experiment on c. We repeat each experiment 30 times and report the average value.

The results for the known input distribution case are shown in Figure 5.2, where the solid lines and shaded areas are the means and standard deviations for 30 trials, respectively. TSPSQ-KNOWN achieved a small regret in the early iterations and outperformed the baselines by finding the nearly optimal solution. Figure 5.3 (left) shows the cumulative regret at the 100th iteration of TSPSQ-UNKNOWN with different values of hyperparameter c for the unknown input distribution case. We observed that the smaller c is better than the larger c in the mean for 30 trials. In particular, TSPSQ-UNKNOWN with $c = 0.12$ worked the best and found the nearly optimal solution, as shown in Figure 5.3 (right) by controlling the balance between exploration and exploitation in the distribution estimation. The result for $c = 0.12$ is competitive with that of TSPSQ-KNOWN for the

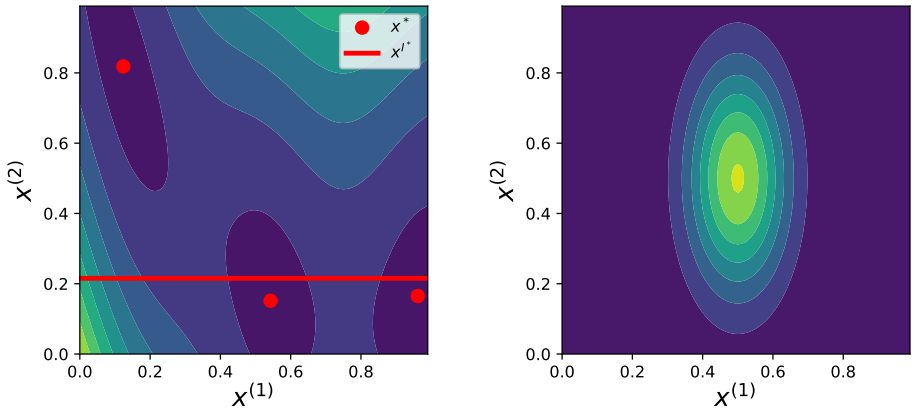


Figure 5.1: Branin-Hoo function (left) and the input probability density function(right).

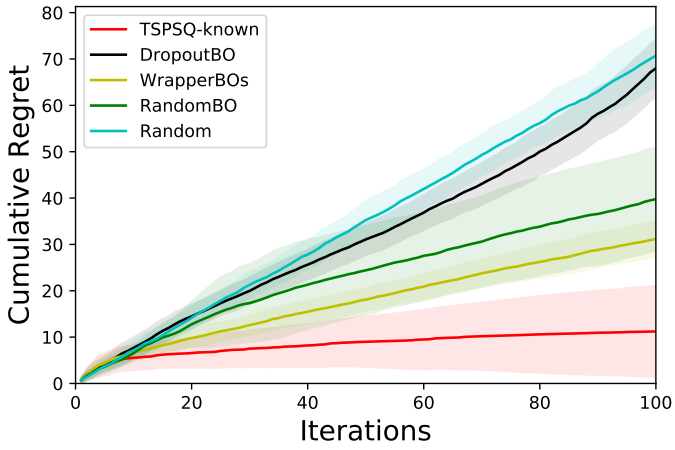


Figure 5.2: Cumulative regret for the Branin-Hoo function with the known input distribution.

known input distribution. The larger c values suffered from the higher regret because they focused on exploration; however, the result is still better than the baselines. The variance for the smaller c values was larger because it focused only on exploitation.

5.5.4 Experiments Using Test Functions

Next, we conduct numerical simulations using two test functions: a cosine mixture function [2] over a two-dimensional domain and the Rosenbrock function [58] over a four-dimensional domain. We model a joint distribution of input variables with a Gaussian

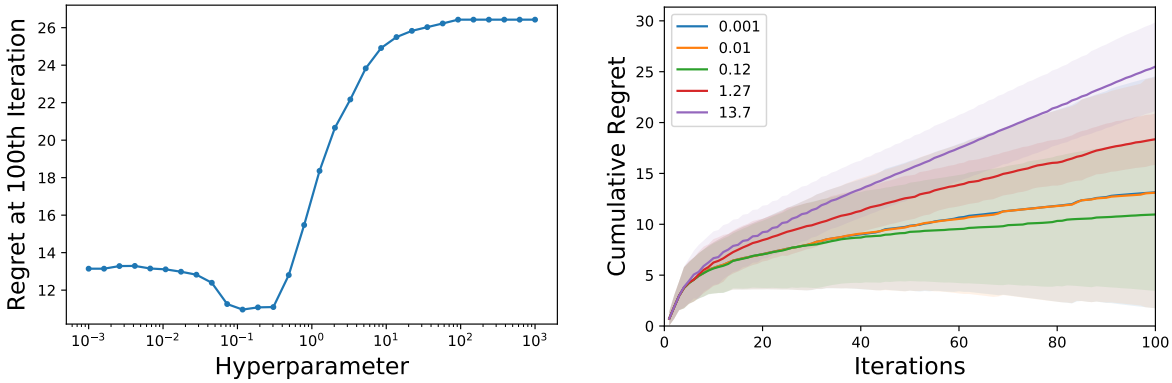


Figure 5.3: Cumulative regret at 100th iteration (left) and the cumulative regret over the iterations (right) of TSPSQ-UNKNOWN with different hyperparameters c for the Branin-Hoo function with the unknown input distribution.

distribution $\mathcal{N}(\mu, \Sigma)$, where we set $\mu = (0.7, 0.7)^\top$, $\Sigma = ((0.01, 0)^\top, (0, 0.05)^\top)$ for the cosine mixture function and $\mu_i = 0.7$, $\Sigma_{i,i} = 0.01, \Sigma_{i,j|i\neq j} = 0, i, j \in [4]$ for the Rosenbrock function. We set the observation variance $\sigma^2 = 10^{-3}$ for both functions, following Hernández-Lobato et al. [27]. The hyperparameters of the kernel functions and the parameter of the observation noise are learned by the type II maximum likelihood estimation every 10 iterations. According to the results in Section 5.5.3, we set $c = 0.12$ for TSPSQ-UNKNOWN for the unknown input distribution case. The other settings are the same as in Section 5.5.3.

Figures 5.4 and 5.5 show the cumulative regret for the known and unknown input distribution cases for the cosine mixture function and the Rosenbrock function, respectively. Note that the performance of the baselines does not change for cases. The proposed algorithms, TSPSQ-KNOWN and TSPSQ-UNKNOWN, outperformed the baselines and achieved the small regret. They finally yielded the nearly optimal solutions, while the baselines increased their cumulative regrets linearly. TSPSQ-UNKNOWN is again competitive with TSPSQ-KNOWN although it is not given the ground-truth input distribution. This result might be because the error in the function estimation is dominant in the overall error than the error in the distribution estimation for TSPSQ-UNKNOWN. The upper bound of TSPSQ-UNKNOWN in Theorem 3 may illustrate this result. In the upper

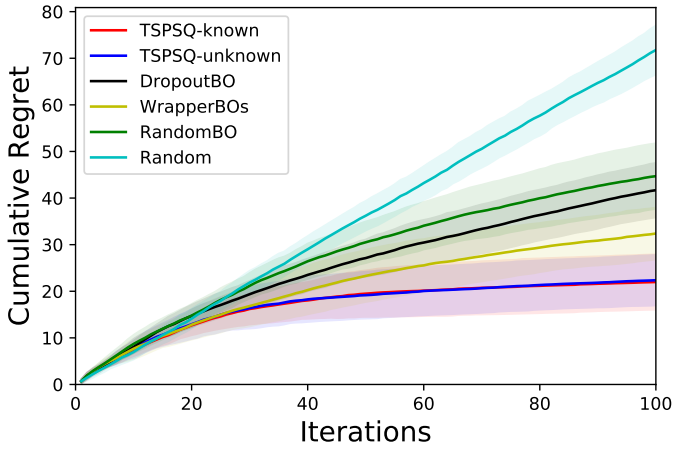


Figure 5.4: Cumulative regret for the cosine mixture function.

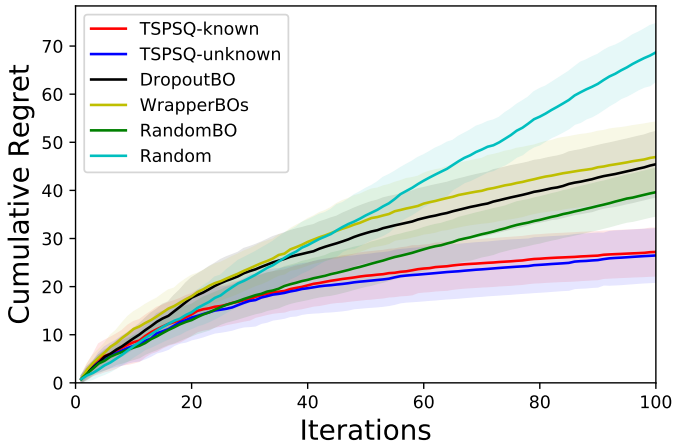


Figure 5.5: Cumulative regret for the Rosenbrock function.

bound, the first term $\sqrt{\gamma_T} \in O\left((\log T)^{(d+1)/2}\right)$ corresponding to the function estimation is larger than the second term $\sqrt{m \log T}$ corresponding to the distribution estimation in growth rate.

5.5.5 Experiments Using Real-World Datasets

Finally, we conduct experiments using two real-world datasets: the airfoil self-noise dataset[1] and the word similarity dataset [68], which is a set of answers by 10 workers

[1]https://archive.ics.uci.edu/ml/datasets/Airfoil+Self-Noise

to 30 word pair similarity tasks in crowdsourcing as shown in Figure 5.6. Using the airfoil self-noise dataset, we aim to minimize airfoil self-noise via 5 input variables including angle of attack and chord length. Using the word similarity dataset, we aim to find good workers whose answers are accurate via 10 input variables. The answer in the dataset is given as an integer from 0 to 10. We use randomly selected 10 answers as the input variables. For the output variable, we calculate the negative mean of the absolute difference between the remaining 20 answers and the true answers so that the good workers have the high output value. We set a family of control variable sets $\mathcal{I}$ as a set of combinations of 2 of 5 input variables for the airfoil self-noise dataset and a set of combinations of 5 of 10 input variables for the word similarity dataset. To reduce the computational cost, we randomly reduce the size of the family to 10 for the word similarity dataset. The discussion on a large $|\mathcal{I}|$ is found in Section 5.3.2. For these datasets, the ground-truth functions and the ground-truth input distributions are unknown and only available through noisy observations. Output data corresponding to the input points we query for is not always available. Therefore, we substitute a posterior mean function of a GP obtained by learning each dataset as each ground-truth function[2]. The observation noise is also set according to the learned parameter. For a proxy for the ground-truth input distributions, we apply kernel density estimation with the Gaussian kernel to the datasets, whose bandwidth is obtained by the median heuristic [20].

Figures 5.7 and 5.8 illustrate the cumulative regret for the known and unknown input distribution cases for the airfoil self-noise function and the word similarity function, respectively. Both TSPSQ-KNOWN and TSPSQ-UNKNOWN performed better than the baselines. In the early iterations, there was no significant difference in the regret of each method, but the proposed algorithms could keep their regrets small after the 20th iteration in Figure 5.7 and after the 30th iteration in Figure 5.8. TSPSQ-UNKNOWN is again competitive with TSPSQ-KNOWN in these experiments, possibly because of the the more difficult function estimation than the distribution estimation, as discussed previously.

[2]This setting is common in real-world data experiments in the BO literature [27].

	boy lad	tool implement	gem jewel	brother monk	lad wizard	brother lad	monk slave	automobile car	forest graveyard	asylum madhouse	...	noon string	magician wizard	shore woodland	rooster voyage	journey voyage
worker #1	10.00	9.00	10.0	9.00	5.00	5.00	0.00	10.00	2.00	10.0	...	2.00	9.00	0.00	2.00	9.00
worker #2	10.00	8.00	9.0	5.00	1.00	3.00	0.00	10.00	0.00	8.0	...	0.00	9.00	0.00	0.00	10.00
worker #3	10.00	9.00	10.0	5.00	0.00	3.00	0.00	10.00	0.00	9.0	...	0.00	10.00	0.00	0.00	9.00
worker #4	10.00	9.00	10.0	5.00	3.00	9.00	5.00	10.00	4.00	10.0	...	0.00	10.00	6.00	0.00	10.00
worker #5	9.00	10.00	9.0	10.00	5.00	6.00	3.00	10.00	6.00	8.0	...	0.00	9.00	3.00	0.00	10.00
worker #6	10.00	8.00	9.0	8.00	4.00	8.00	4.00	10.00	4.00	9.0	...	0.00	10.00	7.00	0.00	10.00
worker #7	10.00	9.00	10.0	9.00	3.00	10.00	3.00	10.00	1.00	10.0	...	0.00	9.00	4.00	0.00	10.00
worker #8	10.00	2.00	10.0	2.00	8.00	2.00	0.00	10.00	0.00	2.0	...	0.00	9.00	9.00	0.00	9.00
worker #9	9.00	9.00	9.0	9.00	0.00	2.00	0.00	9.00	0.00	9.0	...	0.00	9.00	0.00	0.00	9.00
worker #10	10.00	9.00	8.0	1.00	1.00	3.00	0.00	10.00	2.00	8.0	...	0.00	7.00	1.00	0.00	8.00
ground-truth	3.62	2.48	4.0	2.02	1.07	2.17	1.26	3.89	1.24	3.1	...	0.95	3.38	1.09	0.33	3.64

Figure 5.6: Word similarity dataset.

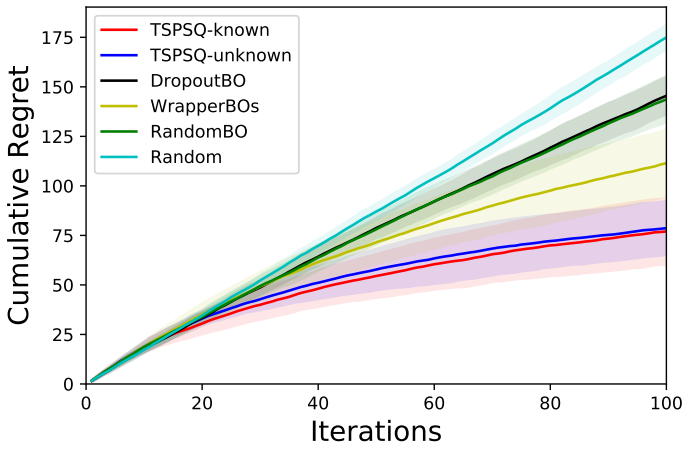


Figure 5.7: Cumulative regret for the airfoil self-noise function.

5.6 Related Work

There is a large body of literature on BO. Here, we primarily focus on variants of the problem settings of BO rather than specific solutions and techniques.

BOPSQ is related to BO where the objective is represented as an expected value of a black-box function f with respect to some random variables. One instance is BO with noisy inputs [55], where an input x is perturbed into $\tilde{x}$ and the objective is defined as $\max_{x \in \mathcal{X}} \mathbb{E}_{\tilde{x}}[f(\tilde{x}) \mid x]$; however, Oliveira et al. [55] handled the usual input queries with input noise but not partially specified queries. Another instance is BO with environmental variables [83], where a black-box function takes control variables x and random variables W, called environmental variables, as input. The objective is defined as the expected value of the black-box function with respect to the environmental variables, $\max_{x \in \mathcal{X}} \mathbb{E}_W[f(x, W)]$. This setting corresponds to BOPSQ when $\mathcal{I} = \{I\}, I \subset [d], 0 < |I| < d$, where x and W are considered control variables and the uncontrol variables, respectively. When $|\mathcal{I}| > 1$ in BOPSQ, the feature selection problem appears, and the learner is required to consider the exploration-exploitation trade-off in the distribution estimation.

The most related work is causal bandit [43], in which a learner considers how to

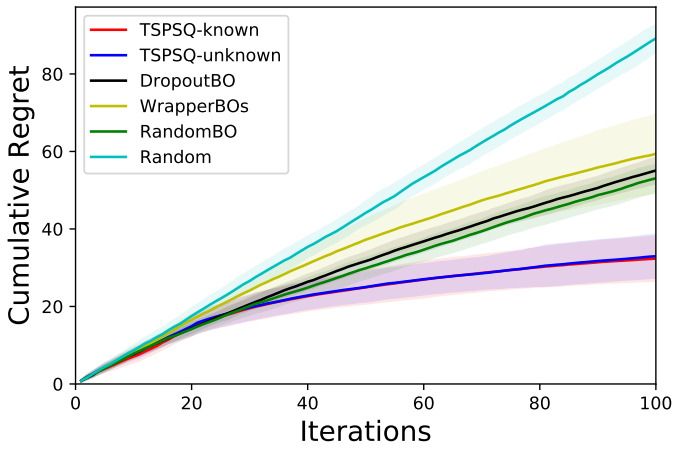


Figure 5.8: Cumulative regret for the word similarity function.

intervene in a causal model. In the problem setting, a causal model is provided as a directed acyclic graph over random variables $X \in \{0,1\}^d$ and $Y \in \mathbb{R}$. In each iteration, the learner specifies values of any subset of X and then observes the remaining subset of X and Y. The objective is to maximize the expectation of Y conditioned on specified values. The primary difference is that BOPSQ deals with infinite input spaces as the Gaussian process bandit, while the causal bandit does finite input spaces as the multi-armed bandit.

Other related works are multi-task BO [72] and contextual Gaussian process bandit optimization [41], where a black-box function returns an output value according to input variables and a task index or context information. The objective is described as $\max_{x \in \mathcal{X}} \sum_w f(x,w)$, where w indicates the task index or context information. Here, x and w can be regarded as control variables and uncontrol variables, respectively, when the family of control variable sets is always fixed to one set, $|\mathcal{I}| = 1$. However, in these settings, neither the input variable selection nor the distribution estimation is performed, and the learner determines x after w is observed.

BOPSQ is also related to the feature selection problem [26], which is the task of selecting a subset of features that are most relevant to an output. In BOPSQ, the selection of a control variable set $I \in \mathcal{I}$ corresponds to feature selection. Chen et al. [12] proposed a method that jointly optimizes the objective and selects features for high-dimensional BO

where it is assumed that only a subset of input variables are associated with a black-box function. Further, Li et al. [45] dealt with high-dimensional BO and their approach is performing optimization only with respect to randomly chosen features. Random values or the values of the input variables with the best function value so far are used as the values of the unchosen features. These methods operate in the standard BO setting, and neither handle partially specified queries nor consider input distribution.

5.7 Conclusion

BO is a promising approach for optimizing an expensive-to-evaluate black-box function. However, the existing BO frameworks can not deal with scenarios where it is difficult to specify all values of input variables. We proposed BOPSQ, which is a novel problem setting that utilizes partially specified queries. In BOPSQ, unlike the standard BO setting, a learner specifies only the values of some input variables, and the values of the unspecified input variables are randomly determined according to a known or unknown distribution. For the known and unknown input distribution cases, we proposed the algorithms based on posterior sampling, TSPSQ-KNOWN and TSPSQ-UNKNOWN. Further, we derived their regret upper bounds that are sublinear for the popular kernels. In the experiments, we demonstrated the effectiveness of the proposed algorithms using the test functions and the real-world datasets.

In BOPSQ, uncontrol variables are sampled from a conditional distribution described by $X^{\bar{I}} \sim P(X^{\bar{I}} \mid X^I = x^I)$. For this sampling, we may consider another scenario: a causal intervention in a directed acyclic graph over random variables [43], described by $X^{\bar{I}} \sim P(X^{\bar{I}} \mid \mathrm{do}(X^I = x^I))$. If the input distribution is neither known nor independent, the intervention scenario would differ from our scenario.

One future direction is to consider the cost of selecting a control variable set [73]. This investigation would be relevant to real-world scenarios, where specifying values of more control variables is more costly. The developments of BOPFQ algorithms for best-

arm identification and parallelized function evaluation should also be investigated in the future. For cases where the input distribution is unknown, TSPSQ-unknown assumes independent input distribution. This assumption may be too strong in some real-world scenarios. Even if the theoretical performance cannot be ensured, it is essential to develop algorithms that relax this assumption.

Chapter 6

Conclusion

In this dissertation, we addressed the learning from small data, which is a difficult problem because of the small amount of information available from the data. The approach in a broader sense is the use of external data other than the original small training data; this approach is further classified into two depending on the type of external data:

- **Auxiliary data**: Auxiliary data is data that can not be directly used for training machine learning models but is useful for that purpose. GD, which is one of the frameworks to exploit auxiliary data, extracts information from auxiliary data via a teacher model to train a student model.

- **Additional data**: Additional data, which is directly used to train machine learning models, is acquired by paying some cost. In the black-box optimization problem, the trade-off between exploration and exploitation needs to be addressed.

We proposed systematic methodologies that explore and exploit useful information to train machine learning models better in three research topics. Those methodologies are based on approaches using such auxiliary data and additional data.

We proposed the framework that improves prediction models by exploring and exploiting auxiliary data in the long-term prediction of small time-series data. The advantage of the framework is that it does not requires a user to prepare auxiliary data, which is

often difficult or costly. The key idea is the exploitation of middle-time data between the input time and the prediction time as auxiliary data based on the property of long-term prediction. Then, a prediction model is trained based on the GD framework using the middle-time data. The experimental results investigating the effects of the middle-time data at different times indicated that middle-time data relatively close to the input time performed well. The experimental results using synthetic data and real-world data showed that the proposed framework improved prediction models especially when the number of input dimensions was high.

Next, we extended change-point detection to an additional data setting, ACPD, where a learner can obtain additional data. We proposed the general meta-algorithm that can be applied to different types of data and change-points. Our idea was considering ACPD as a black-box optimization problem of an unknown change score function based on an existing change-point detection approach. A BO technique was applied to change scores computed by a change-point detection technique to determine the data to be acquired next. The formulation allows a user to change a kernel function and change score definition based on the types of data and change-point of interest. The proposed meta-algorithm outperformed the baselines by balancing exploration and exploitation of information in the experiments using synthetic data and real-world data such as the phase transition data and the seafloor depth data.

ACPD and the standard BO problems assumed that the input query is deterministic: a learner can specify the values of all input variables. We considered the novel BO problem called BOPSQ where the input query is random. In particular, a learner can specify the values of some input variables, and the values of unspecified variables are randomly determined according to a known or unknown input distribution. We proposed the two algorithms based on Thompson sampling for scenarios where the input distribution is known and unknown. The algorithm for the unknown input distribution case employs a multi-armed bandit approach to appropriately explore the input distribution. We derived their regret upper bounds, which are sublinear for the popular kernel functions. We

demonstrated the effectiveness of the proposed algorithms using the synthetic function and the real-world datasets.

From a broader perspective, the underlying problem is the justification of assumptions for the use of methods in real-world scenarios. In the approach using auxiliary data, the crucial problem is checking if auxiliary data is truly effective for the target task. We introduced that auxiliary data is possibly helpful for training machine learning models. However, when it is not relevant to the task, it would rather deteriorate the model. Further, the experimental results in Chapter 3 showed that the middle-time data was not effective for the easy problems. Existing approaches using auxiliary data assume that auxiliary data has a relationship to the target data, which can be represented as a graphical model. Causal discovery methods [29] that estimate the graphical model of random variables can be possibly used to check if auxiliary data works according to the assumption. However, causal discovery with small data itself can not be easily solved. In the approach using additional data, model misspecification is a large problem. Although the regret upper bound of a BO method is sublinear, it is guaranteed only if the unknown function follows a GP: smooth and no jump points. However, real-world functions of interest often do not meet the assumptions. When a model misspecification occurs, a BO method may perform worse than random sampling. A goodness of fit test [3] can determine if the specified model fits data well. However, like causal discovery, the goodness of fit test with small data is itself a difficult problem and requires another assumption. According to the above consideration, there may not exist a general solution without any assumption. However, from an application point of view, it may be enough that methods work well in real-world scenarios, and approaches such as transfer learning [57] and meta-learning [17] aim at this by generalizing across different tasks. Incorporating other types of knowledge such as physical systems into models may be another promising approach.

Publications

Journal Articles

[J1] Shogo Hayashi, Akira Tanimoto, and Hisashi Kashima. Long-term prediction of small time-series data using generalized distillation. *The Japanese Society for Artificial Intelligence*, 35(5):B-K33_1-9, 2020.

[J2] Shogo Hayashi, Yoshinobu Kawahara, and Hisashi Kashima. Active Change-Point Detection. *The Japanese Society for Artificial Intelligence*, 35(5):E-JA10_1-10, 2020.

Peer-reviewed Conference Proceedings

[C1] Shogo Hayashi, Akira Tanimoto, and Hisashi Kashima. Long-Term Prediction of Small Time-Series Data Using Generalized Distillation. In *Proceedings of International Joint Conference on Neural Networks*, 2019.

[C2] Shogo Hayashi, Yoshinobu Kawahara, and Hisashi Kashima. Active change-point detection, In *Proceedings of the 10th Asian Conference on Machine Learning*, 2019.

Bibliography

[1] H. Akaike. Fitting autoregressive models for prediction. *Annals of the institute of Statistical Mathematics*, 21(1):243–247, 1969.

[2] B. S. Anderson, A. W. Moore, and D. Cohn. A nonparametric approach to noisy and costly optimization. In *Proceedings of the 17th International Conference on Machine Learning*, pages 17–24, 2000.

[3] T. W. Anderson and D. A. Darling. A test of goodness of fit. *Journal of the American Statistical Association*, 49(268):765–769, 1954.

[4] M. A. Anisimov. *Critical phenomena in liquids and liquid crystals*. Gordon and Breach Science Publishers, 1991.

[5] S. Ao, X. Li, and C. X. Ling. Fast generalized distillation for semi-supervised domain adaptation. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, pages 1719–1725, 2017.

[6] P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2-3):235–256, 2002.

[7] M. Basseville and I. V. Nikiforov. *Detection of abrupt changes: theory and application*. Prentice Hall, 1993.

[8] S. Basu and G. Michailidis. Regularized estimation in sparse high-dimensional time series models. *Annals of Statistics*, 43(4):1535–1567, 2015.

[9] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[10] L. Cao and F. E. H. Tay. Support vector machine with adaptive parameters in financial time series forecasting. *IEEE Transactions on Neural Networks*, 14(6): 1506–1518, 2003.

[11] K. Chaloner and I. Verdinelli. Bayesian experimental design: A review. *Statistical Science*, 10(3):273–304, 1995.

[12] B. Chen, R. M. Castro, and A. Krause. Joint optimization and variable selection of high-dimensional Gaussian processes. In *Proceedings of the 29th International Conference on Machine Learning*, 2012.

[13] Y. Chen, B. Yang, and J. Dong. Time-series prediction using a local linear wavelet neural network. *Neurocomputing*, 69(4-6):449–465, 2006.

[14] H. Cheng, P. Tan, J. Gao, and J. Scripps. Multistep-ahead time series prediction. In *Proceedings of the 10th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, pages 765–774, 2006.

[15] S. Das, W.-K. Wong, T. Dietterich, A. Fern, and A. Emmott. Incorporating expert feedback into active anomaly discovery. In *Proceedings of the 16th IEEE International Conference on Data Mining*, pages 853–858, 2016.

[16] A. Dvoretzky, J. Kiefer, and J. Wolfowitz. Asymptotic minimax character of the sample distribution function and of the classical multinomial estimator. *The Annals of Mathematical Statistics*, 27(3):642–669, 1956.

[17] C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1126–1135, 2017.

[18] P. Fryzlewicz. Wild binary segmentation for multiple change-point detection. *The Annals of Statistics*, 42(6):2243–2281, 2014.

[19] J. a. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia. A survey on concept drift adaptation. *ACM Computing Surveys*, 46(4):44:1–44:37, 2014.

[20] D. Garreau, W. Jitkrittum, and M. Kanagawa. Large sample analysis of the median heuristic. *arXiv preprint arXiv:1707.07269*, 2017.

[21] I. Gijbels, P. Hall, and A. Kneip. On the estimation of jump points in smooth curves. *Annals of the Institute of Statistical Mathematics*, 51(2):231–251, 1999.

[22] A. Grigorievskiy, Y. Miche, A. Ventelä, E. Séverin, and A. Lendasse. Long-term time series prediction using OP-ELM. *Neural Networks*, 51:50–56, 2014.

[23] H. Guo, W. Pedrycz, and X. Liu. Hidden markov models based approaches to long-term prediction for granular time series. *IEEE Transactions Fuzzy Systems*, 26(5): 2807–2817, 2018.

[24] V. Guralnik and J. Srivastava. Event detection from time series data. In *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 33–42, 1999.

[25] F. Gustafsson. *Adaptive filtering and change detection*. Wiley, 2000.

[26] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of machine learning research*, 3:1157–1182, 2003.

[27] J. M. Hernández-Lobato, M. W. Hoffman, and Z. Ghahramani. Predictive entropy search for efficient global optimization of black-box functions. In *Advances in Neural Information Processing Systems 27*, pages 918–926, 2014.

[28] G. E. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[29] P. O. Hoyer, D. Janzing, J. M. Mooij, J. Peters, and B. Schölkopf. Nonlinear causal discovery with additive noise models. In *Advances in Neural Information Processing Systems 21, Proceedings of the 22nd Annual Conference on Neural Information Processing Systems*, pages 689–696, 2008.

[30] C. M. Hurvich and C.-L. Tsai. Regression and time series model selection in small samples. *Biometrika*, 76(2):297–307, 1989.

[31] T. Idé, D. T. Phan, and J. Kalagnanam. Change detection using directional statistics. In S. Kambhampati, editor, *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, pages 1613–1619, 2016.

[32] R. Jonschkowski, S. Höfer, and O. Brock. Patterns for learning with side information. *arXiv preprint arXiv:1511.06429*, 2015.

[33] N. M. Joy, S. R. Kothinti, S. Umesh, and B. Abraham. Generalized distillation framework for speaker normalization. In *The 18th Annual Conference of the International Speech Communication Association*, pages 739–743, 2017.

[34] N. M. Joy, S. R. Kothinti, and S. Umesh. FMLLR speaker normalization with i-vector: In pseudo-FMLLR and distillation framework. *IEEE ACM Transactions on Audio, Speech, and Language Processing*, 26(4):797–805, 2018.

[35] J. M. P. M. Jr. and G. A. Barreto. Long-term time series prediction with the NARX network: An empirical evaluation. *Neurocomputing*, 71(16-18):3335–3343, 2008.

[36] K. Kandasamy, J. G. Schneider, and B. Póczos. High dimensional Bayesian optimisation and bandits via additive models. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 295–304, 2015.

[37] Y. Kawahara, T. Yairi, and K. Machida. Change-point detection in time-series data based on subspace identification. In *Proceedings of the 7th IEEE International Conference on Data Mining*, pages 559–564, 2007.

[38] E. J. Keogh, S. Chu, D. M. Hart, and M. J. Pazzani. An online algorithm for segmenting time series. In *Proceedings of the 1st IEEE International Conference on Data Mining*, pages 289–296, 2001.

[39] K. Korovina, S. Xu, K. Kandasamy, W. Neiswanger, B. Póczos, J. Schneider, and E. P. Xing. Chembo: Bayesian optimization of small organic molecules with synthesizable recommendations. In *The 23rd International Conference on Artificial Intelligence and Statistics*, pages 3393–3403, 2020.

[40] Y. Kou, C.-T. Lu, and D. Chen. Spatial weighted outlier detection. In *Proceedings of the 2006 SIAM International Conference on Data Mining*, pages 614–618, 2006.

[41] A. Krause and C. S. Ong. Contextual Gaussian process bandit optimization. In *Advances in Neural Information Processing Systems 24*, pages 2447–2455, 2011.

[42] M. Längkvist, L. Karlsson, and A. Loutfi. A review of unsupervised feature learning and deep learning for time-series modeling. *Pattern Recognition Letters*, 42:11–24, 2014.

[43] F. Lattimore, T. Lattimore, and M. D. Reid. Causal bandits: Learning good interventions via causal inference. In *Advances in Neural Information Processing Systems 29*, pages 1181–1189, 2016.

[44] D. D. Lewis and W. A. Gale. A sequential algorithm for training text classifiers. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3–12, 1994.

[45] C. Li, S. Gupta, S. Rana, V. Nguyen, S. Venkatesh, and A. Shilton. High dimensional Bayesian optimization using dropout. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 2096–2102, 2017.

[46] H. Li, B. Zhao, and A. Fuxman. The wisdom of minority: discovering and targeting

the right group of workers for crowdsourcing. In *23rd International World Wide Web Conference*, pages 165–176, 2014.

[47] S. Liu, M. Yamada, N. Collier, and M. Sugiyama. Change-point detection in time-series data by relative density-ratio estimation. *Neural Networks*, 43:72–83, 2013.

[48] D. Lopez-Paz, L. Bottou, B. Schölkopf, and V. Vapnik. Unifying distillation and privileged information. *arXiv preprint arXiv:1511.03643*, 2015.

[49] K. Markov and T. Matsui. Robust speech recognition using generalized distillation framework. In *The 17th Annual Conference of the International Speech Communication Association*, pages 2364–2368, 2016.

[50] P. Massart. The tight constant in the Dvoretzky-Kiefer-Wolfowitz inequality. *The Annals of Probability*, 18(3):1269–1283, 1990.

[51] J. Mockus, V. Tiesis, and A. Zilinskas. *Toward Global Optimization*, volume 2, chapter The application of Bayesian Methods for Seeking the Extremum, pages 117–129. Elsevier, 1978.

[52] K. Müller, A. J. Smola, G. Rätsch, B. Schölkopf, J. Kohlmorgen, and V. Vapnik. Predicting time series with support vector machines. In *Proceedings of the 7th International Conference on Artificial Neural Networks*, pages 999–1004, 1997.

[53] M. Mutny and A. Krause. Efficient high dimensional Bayesian optimization with additivity and quadrature Fourier features. In *Advances in Neural Information Processing Systems 31*, pages 9019–9030, 2018.

[54] J. F. L. Oliveira and T. B. Ludermir. Iterative arima-multiple support vector regression models for long term time series prediction. In *The 22th European Symposium on Artificial Neural Networks*, 2014.

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

[56] N. R. Pal and S. K. Pal. A review on image segmentation techniques. *Pattern Recognition*, 26(9):1277 – 1294, 1993.

[57] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.

[58] V. Picheny, T. Wagner, and D. Ginsbourger. A benchmark of kriging-based infill criteria for noisy optimization. *Structural and Multidisciplinary Optimization*, 48(3): 607–626, 2013.

[59] A. Rahimi and B. Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems 20*, pages 1177–1184, 2007.

[60] C. E. Rasmussen and C. K. I. Williams. *Gaussian processes for machine learning*. MIT Press, 2006.

[61] W. Rudin. *Fourier Analysis on Groups*. John Wiley & Sons, 2011.

[62] D. Russo and B. V. Roy. Learning to optimize via posterior sampling. *Mathematics of Operations Research*, 39(4):1221–1243, 2014.

[63] N. Sebastián, M. R. De La Fuente, D. O. López, M. A. Pérez-Jubindo, J. Salud, S. Diez-Berart, and M. B. Ros. Dielectric and thermodynamic study on the liquid crystal dimer α-(4-Cyanobiphenyl-4'-oxy)-ω-(1-pyreniminebenzylidene-4'-oxy) undecane (CBO11o·py). *The Journal of Physical Chemistry B*, 115(32):9766–9775, 2011.

[64] B. Settles. Active learning literature survey. Technical report, University of Wisconsin, Madison, 2010.

[65] B. Settles. *Active learning*. Morgan & Claypool Publishers, 2012.

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

[67] J. Snoek, H. Larochelle, and R. P. Adams. Practical Bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems 25*, pages 2960–2968, 2012.

[68] R. Snow, B. O'Connor, D. Jurafsky, and A. Y. Ng. Cheap and fast - but is it good? evaluating non-expert annotations for natural language tasks. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 254–263, 2008.

[69] A. Sorjamaa, J. Hao, N. Reyhani, Y. Ji, and A. Lendasse. Methodology for long-term prediction of time series. *Neurocomputing*, 70(16-18):2861–2869, 2007.

[70] N. Srinivas, A. Krause, S. M. Kakade, and M. W. Seeger. Information-theoretic regret bounds for Gaussian process optimization in the bandit setting. *IEEE Transactions on Information Theory*, 58(5):3250–3265, 2012.

[71] S. Sun. A survey of multi-view machine learning. *Neural Computing and Applications*, 23(7-8):2031–2038, 2013.

[72] K. Swersky, J. Snoek, and R. P. Adams. Multi-task Bayesian optimization. In *Advances in Neural Information Processing Systems 26*, pages 2004–2012, 2013.

[73] L. Tran-Thanh, A. C. Chapman, A. Rogers, and N. R. Jennings. Knapsack based optimal policies for budget-limited multi-armed bandits. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence*, 2012.

[74] C. Truong, L. Oudre, and N. Vayatis. A review of change point detection methods. *arXiv preprint arXiv:1801.00718*, 2018.

[75] V. Vapnik. *Statistical learning theory. 1998*, volume 3. Wiley, New York, 1998.

[76] V. Vapnik and R. Izmailov. Learning using privileged information: similarity control and knowledge transfer. *Journal of Machine Learning Research*, 16:2023–2049, 2015.

[77] V. Vapnik and A. Vashist. A new learning paradigm: Learning using privileged information. *Neural Networks*, 22(5-6):544–557, 2009.

[78] A. Venkatraman, M. Hebert, and J. A. Bagnell. Improving multi-step prediction of learned time series models. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, pages 3024–3030, 2015.

[79] A. Vezhnevets, J. M. Buhmann, and V. Ferrari. Active learning for semantic segmentation with expected change. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3162–3169, 2012.

[80] I. Žliobaitė, A. Bifet, B. Pfahringer, and G. Holmes. Active learning with drifting streaming data. *IEEE Transactions on Neural Networks and Learning Systems*, 25 (1):27–39, 2014.

[81] D. Wang, C. Yan, S. Shan, and X. Chen. Active learning for interactive segmentation with expected confidence change. In *The 11th Asian Conference on Computer Vision*, pages 790–802. Springer, 2013.

[82] Y. Wang, A. Chakrabarti, D. Sivakoff, and S. Parthasarathy. Fast change point detection on dynamic social networks. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 2992–2998, 2017.

[83] B. J. Williams, T. J. Santner, and W. I. Notz. Sequential design of computer experiments to minimize integrated response functions. *Statistica Sinica*, 10(4):1133–1152, 2000.

[84] K. Yamanishi, J. Takeuchi, G. J. Williams, and P. Milne. On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms. *Data Mining and Knowledge Discovery*, 8(3):275–300, 2004.

[85] Y.-C. Yao. Estimating the number of change-points via schwarz' criterion. *Statistics & Probability Letters*, 6(3):181 – 189, 1988.

[86] X. J. Zhu. Semi-supervised learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2005.