Proceedings of Machine Learning Research 101:1017–1032, 2019 ACML 2019

Active Change-Point Detection

Shogo Hayashi HAYASHI@ML.IST.I.KYOTO-U.AC.JP
Graduate School of Informatics, Kyoto University

Yoshinobu Kawahara KAWAHARA@IMI.KYUSHU-U.AC.JP
Institute of Mathematics for Industry, Kyushu University
RIKEN Center for AIP

Hisashi Kashima KASHIMA@I.KYOTO-U.AC.JP
Graduate School of Informatics, Kyoto University
RIKEN Center for AIP

Editors: Wee Sun Lee and Taiji Suzuki

Abstract

We introduce Active Change-Point Detection (ACPD), a novel active learning problem for efficient change-point detection in situations where the cost of data acquisition is expensive. At each round of ACPD, the task is to adaptively determine the next input, in order to detect the change-point in a black-box expensive-to-evaluate function, with as few evaluations as possible. We propose a novel framework that can be generalized for different types of data and change-points, by utilizing an existing change-point detection method to compute change scores and a Bayesian optimization method to determine the next input. We demonstrate the efficiency of our proposed framework in different settings of datasets and change-points, using synthetic data and real-world data, such as material science data and seafloor depth data.

Keywords: change-point detection, Bayesian optimization

1. Introduction

The problem of detecting abrupt changes in data is called change-point detection (Basseville and Nikiforov, 1993; Gustafsson, 2000). It has been enthusiastically studied in data mining and industry, covering a broad range of data types, such as sensor data (Idé et al., 2016) and dynamic network data (Wang et al., 2017), among others. Its applications include, for example, fault detection (Kawahara et al., 2007), network-intrusion detection (Yamanishi et al., 2004), and trend change detection (Liu et al., 2013).

Typical change-point detection problems assume that parts of time series data are incrementally observed in an online manner, or that the entire data is given at once in a batch manner. In both settings, they passively observe the data and do not consider the cost of data acquisition. However, when the data acquisition cost is high and unignorable, we need to choose informative data points in an interactive manner for saving the costs.

One motivating example is found in material science. Imagine a physical experiment that attempts to detect a phase transition temperature of a material (Figures 1, 2 and 3). Phase transitions are sudden changes in physical properties (e.g., density, energy, electric resistance, and specific heat) and phases (e.g., gel-sol, solid-liquid, and nematic-isotropic) at particular temperatures. Finding phase transition temperatures is crucial for developing new materials. They are usually examined via real experiments and simulations, wherein we set a material at a particular temperature and observe its physical property by consuming some financial or temporal resources. Another example is found in geoscience. Studying the geography of seafloor which has rugged landscapes (Figure 8(a)) is important for understanding ocean currents and other phenomena. Measuring the depth of the sea requires considerable amount of resources, and we want to reduce the number of locations to make measurements. In both of the examples, it is desirable to save data acquisition costs by interactively choosing the conditions or locations of measurements based on past observations, rather than measuring randomly or thoroughly.

In this paper, we consider Active Change-Point Detection (ACPD), a novel active learning problem of change-point detection with data acquisition cost. 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. Therefore, we aim to determine an effective sequence of input queries to find the change-point using as few queries as possible. 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 and latitude) and the depth. Note that ACPD does not consider the input is time.

We propose a simple and general solution to ACPD, which does not rely on the underlying data structures or the definitions of change-points. Our solution is a meta-algorithm based on an idea that we regard ACPD as black-box optimization of change scores. It reuses an existing change-point detection algorithm to compute change scores from data. It also employs a Bayesian optimization 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 of different types of data and change-points, demonstrate the efficiency of the proposed framework.

The main contributions of this paper are summarized as follows:

- Proposal of ACPD, a novel active learning problem for efficient change-point detection in situations where the cost of data acquisition is expensive so that we need to determine the next input to be evaluated in an interactive manner.

- A general solution to the ACPD problem, which is a meta-algorithm exploiting an existing change detection method and a Bayesian optimization technique, and is applicable to a variety types of data and change-points.

- Empirical supports for ACPD framework using synthetic data sets and real-world data sets.

2. Problem Settings

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

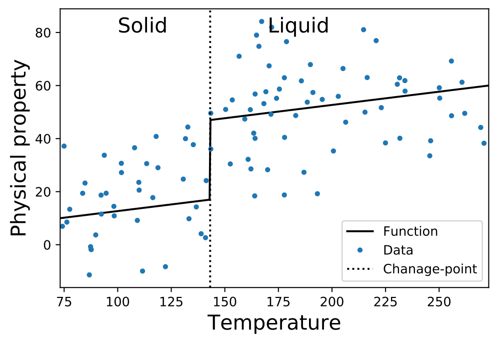


Figure 1: First-order phase transition from solid to liquid state at 143.15 and randomly-sampled 100 observations.

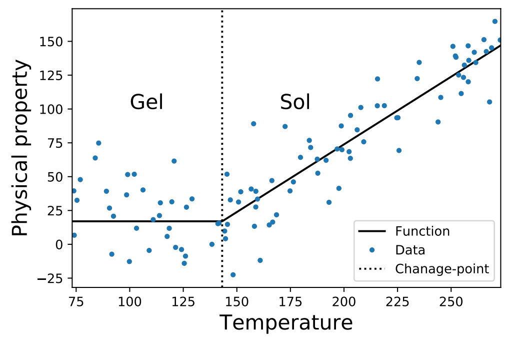


Figure 2: Second-order phase transition from gel to sol at 143.15 and randomly-sampled 100 observations.

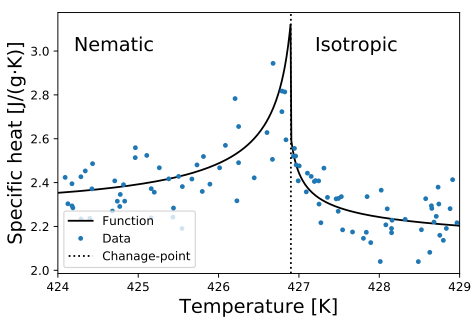


Figure 3: Nematic-isotropic phase transition of the CBO11O material at 426.9 [K] (Sebastián et al., 2011) and randomly-sampled 100 observations.

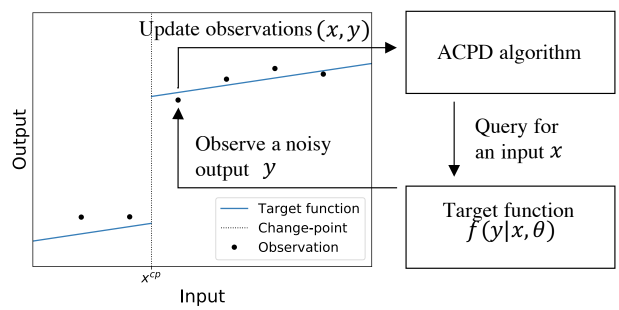


Figure 4: An overview of the active change-point detection procedure. At every iteration, an ACPD algorithm queries a black-box target function for an input, observes a noisy output, and updates its internal models to determine the next query.

2.1. Change-Point Detection

There exist a variety of problem settings of change-point detection up to definitions of data and change-points. For simplicity, we review the standard (passive) problem setting of 1-dimensional change-point detection with a change-point in an offline setting (Basseville and Nikiforov, 1993; Gijbels et al., 1999). Suppose we have N observations $\mathcal{D} = (X, Y) = \{(x_i, y_i)\}_{i=1}^N$, where the input $x_i \in \mathcal{X}$ (we assume $\mathcal{X}$ is an interval in $\mathbb{R}$) and the output $y_i \in \mathbb{R}$ are associated by an unknown function $f(y_i|x_i, \theta)$, which is parameterized by a piecewise-constant parameter θ. Here, we define a change-point $x^{\mathrm{cp}} \in X$ as the point where the model parameter "suddenly" changes from $\theta = \theta_1$ to $\theta = \theta_2$. The typical settings deal with temporal data, that is, x_i corresponds to a time index, which we do not assume in this paper. For example, 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 (with some noise) f (Figures 1, 2 and 3).

In change-point detection problems, the goal is to find the change-point $x^{\mathrm{cp}} \in X$. A typical approach is to model a *change score function* $s_{\mathcal{D}} \colon X \to \mathbb{R}$, which quantifies how f

"changes" over the inputs. The change-point is estimated as a point that maximizes the change score:

$$\hat{x}^{\text{cp}} = \arg\max_{x \in X} s_{\mathcal{D}}(x). \tag{1}$$

The change score function is designed in such a way 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

$$s_{\mathcal{D}}(x) = \max_{\theta_1, \theta_2} \ln \left[\prod_{j | x_j < x} P(y_j \mid x_j, \theta_1) \prod_{k | x \le x_k} P(y_k \mid x_k, \theta_2) \right]. \tag{2}$$

Typically, additive Gaussian noise is assumed: $P(y \mid x, \theta) = \mathcal{N}(y \mid f(x), \sigma^2)$. The change score is computed according to how much the model with the parameters fits the data.

2.2. Active Change-Point Detection (ACPD) Problem

In contrast to the passive change-point detection problem, where we have no control over the inputs to the target system f, the ACPD problem allows us 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^{cp} by making queries to the target function f in an iterative manner. At each iteration, we first determine the next input query based on the past observations, we give an input query $x \in \mathcal{X}$ to the target function, and then observe the corresponding output y. We usually have a limited budget B, which is the maximum number of queries we can make in total; therefore, an ACPD algorithm is required to suggest an effective sequence of input queries to find the change-point based on the past observations. An overview of the ACPD procedure is illustrated in Figure 4.

Note that, for simplicity, we assume one-dimensional continuous input and output variables, and there exists only one change-point as a point with a sudden change in our problem setting; however, the problem setting can be extended to other types of inputs, outputs, and change-points, for example, multi-dimensional inputs, outputs and multiple change-point cases, which we will show in the experimental part.

3. Proposed Method

There are many possible definitions of change-points and change score functions depending on the 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 method, which is applicable to a wide variety of data and change-points.

Our key idea is to see the ACPD problem as a black-box optimization problem of a change score function. As a summary of the meta-algorithm, we utilize an existing change-point detection method to compute change scores and then perform Bayesian optimization (Mockus et al., 1978; Snoek et al., 2012) on the change scores to decide the next input that would maximize the change score. At each iteration, we use a change-point detection method to obtain the change scores S for the data points X that we have observed so far.

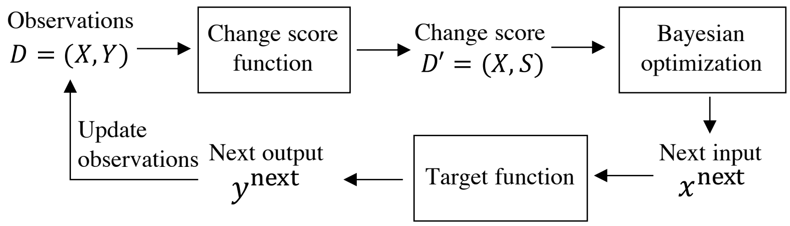


Figure 5: Our proposed Meta-ACPD method. For the current input-output observations $\mathcal{D}$ of the target system, it exploits an existing change score function from a change-point detection algorithm to obtain a set of change scores $\mathcal{D}'$. A Bayesian optimization algorithm determines the next query input x^{next}. The observations $\mathcal{D}$ and the models are updated according to the response from the target system. The above procedure is iterated until the budget B runs out.

We next estimate the change scores over the input space using a Gaussian process with the computed change scores, and determine the next query input x^{next} such that it would maximize the change score using an acquisition function. We iterate this procedure until the given budget runs out, and then output 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 5 and Algorithm 1.

We describe the meta-algorithm in detail below. In the setting of 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 try to find the change-point in a domain $\mathcal{X}$, such as $\mathbb{R}$. If we had infinite number of observations $\mathcal{D}^\infty = (\mathcal{X}, Y)$ over the entire domain $\mathcal{X}$, we could estimate the change-point in the same way as the passive offline change-point detection, by finding the maximizer in a set of change scores. The key idea is to regard the change scores over the entire the domain as a function $s : \mathcal{X} \to \mathbb{R}$ and see the ACPD problem as a black-box optimization problem of the unknown change score function s using a finite number of observations, to estimate the maximizer of the change score function as the change-point. However, in addition to the fact that observations are finite and might be noisy, the change scores depend on the selection of the observations; different sets of observations produce different change scores. To take the uncertainty of the computed change scores in consideration, we model the change score function using a probabilistic model, a Gaussian process, which enables us to utilize Bayesian optimization to determine the next input for finding the maximum change score.

Given N observations $\mathcal{D} = (X, Y) = \{(x_i, y_i)\}_{i=1}^N$, we apply a change score function of a change-point detection algorithm to $\mathcal{D}$ in order to obtain a set of change scores $\mathcal{D}'$ as

$$\mathcal{D}' = \{(x_i, s_{\mathcal{D},i}) \mid s_{\mathcal{D},i} = s_{\mathcal{D}}(x_i)\}_{i=1}^N = (X, S_{\mathcal{D}}). \tag{3}$$

We assume that a black-box change score function s follows a Gaussian process (Rasmussen and Williams, 2006), that is, $s_{\mathcal{D}}$ follows a Gaussian distribution with mean $\mu(x)$ and variance $\sigma^2(x)$:

$$s_{\mathcal{D}} \sim \mathcal{N}\left(\mu(x), \sigma^2(x)\right). \tag{4}$$

Algorithm 1 Meta-ACPD

Input:

Change-point detection algorithm and its change score function: s

Acquisition function: a

Query budget: B

Initial set of observations: $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$

Output: Change-point estimate $\hat{x}^{\mathrm{cp}}$

```
for t = 1, 2, ..., B do
    Apply s to D and obtain change scores D′ (Eq. (3))
    Determine the next input x^next using an acquisition function a (Eq. (7))
    Observe the output y^next by evaluating the target function at x^next
    Update the observations D ← D ∪ {(x^next, y^next)}
end for
return The final change-point estimate by the change-point detection algorithm x̂^cp
```

Given the computed change scores $\mathcal{D}'$, the posterior mean $\mu_{\mathcal{D}'}(x)$ and variance $\sigma^2_{\mathcal{D}'}(x)$ for an input x are respectively estimated from $\mathcal{D}'$ as

$$\mu_{\mathcal{D}'}(x) = k(x, X)(K + \lambda^2 I)^{-1} S_{\mathcal{D}}, \tag{5}$$

$$\sigma^2_{\mathcal{D}'}(x) = k(x, x) - k(x, X)(K + \lambda^2 I)^{-1} k(X, x), \tag{6}$$

where $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a positive definite function, K is an $N \times N$ Gram matrix whose (i, j)-th entry is $k(x_i, x_j)$, and λ^2 is the variance of the output noise.

Next, we design the next input using Bayesian optimization. In Bayesian optimization, after the Gaussian process regression, an acquisition function $a\colon \mathcal{X} \to \mathbb{R}$, which quantifies how much an input point should be evaluated, is used to determine the next input point x^{next}. Using the Gaussian process and an arbitrary acquisition function $a(x)$, we determine the next input x^{next} that gives the maximum value of the acquisition function:

$$x^{\mathrm{next}} = \arg\max_{x \in \mathcal{X}} a(x). \tag{7}$$

One possible choice for the acquisition function is the upper confidence bound algorithm (Srinivas et al., 2010): $a(x) = \mu_{\mathcal{D}'}(x) + \kappa\sigma_{\mathcal{D}'}(x)$, where κ is a hyperparameter that balances exploration and exploitation. Another choice is the expected improvement (Mockus et al., 1978), $a(x) = \sigma(x)\left(\gamma(x)\Phi\left(\gamma(x)\right) + \mathcal{N}(\Phi\left(\gamma(x)\right) \mid 0, 1)\right)$, where $\gamma(x) = \frac{\mu(x) - y^{best}}{\sigma(x)}$, y^{best} is the current greatest observed output, and $\Phi(\cdot)$ denotes the cumulative distribution function of the standard normal.

By evaluating the target function at the next input, we observe the corresponding output y^{next}. The above iteration terminates when the query budget runs out or some particular condition is satisfied. Consequently, we estimate the change-point using the change-point detection algorithm.

One of the significant advantages of the proposed method is that it does not depend on the target data types and the definitions of the change-points, because it is a meta-algorithm utilizing an existing change-point detection algorithm to compute one-dimensional change scores. It would work for other types of inputs, outputs, and changes, such as multi-dimensional input-output, noise-level changes, and multiple change-points, as long as the underlying change-point algorithm can compute the change scores and we can define an appropriate kernel for the data structure.

However, the change scores, which varies depending on data, might cause biases in the next input and therefore fall into a wrong solution, especially in the early stages with a small number of data. In practice, Bayesian optimization could mitigate the problem, because it explores uniformly in the domain in the early stages. We will discuss this problem in the experimental part from a perspective of exploration and exploitation.

4. Experiments

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

4.1. Target Functions

4.1.1. Functions with one change-point

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

- a non-continuous piecewise-linear function $\mathrm{PT}_{\mathrm{bias}}$: $f(x) = 0.1x + 30$ if $x < 143.15$, and $0.1x + 60$ otherwise (Figure 1), and
- a continuous piecewise-linear function $\mathrm{PT}_{\mathrm{slope}}$: $f(x) = 0.1x + 30$ if $x < 143.15$, and $x + 147$ otherwise (Figure 2).

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

We also use a multiple-output function MO, that is a piecewise-linear function with five-dimensional outputs. MO is defined as $f(x) = ax+b$ if $x < -130$, and $f(x) = ax+b+50$ otherwise, 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 3). The function is given as

$$\mathrm{NI}: f(x) = \begin{cases} 2.13 - 2.99\,(x/427.03 - 1) + 0.0162\,|x/427.03 - 1|^{-0.51} & (424 \leq x < 426.9) \\ 2.13 - 2.99\,(x/426.82 - 1) + 0.006\,|x/426.82 - 1|^{-0.51} & (426.9 \leq x \leq 429) \end{cases}, \tag{8}$$

that is a regression result in a real experiment (Anisimov, 1991; Sebastián et al., 2011).

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

4.1.2. 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. The function is defined as $f(x) = 9.34$ if $x < -143.5$, 0.94 if $-143.5 \leq x < -77.7$, 9.45 if $-77.7 \leq x < -41.9$, and 4.30 otherwise. Gaussian noise $\mathcal{N}(0, 1^2)$ is added to its output. Figure 7 shows the function and its noisy observations.

4.1.3. 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 bottom surveys. We use a seafloor depth dataset[1] for the area around Hokkaido, Japan, provided by Cabinet Office, Government of Japan. The dataset is 450m-mesh data of 840×780 cells, and we down-sample it into 280×260 for computational resource, as shown in Figure 8(a), where the land area is colored in white. The two-dimensional location (longitude and latitude) is the input, and the depth is the output.

4.2. Comparing Method

To the best of our knowledge, there is no existing method directly addressing the ACPD problems. Hence, we compare the proposed meta-algorithm with two naive baselines: a random search and an 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. The 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 $\epsilon \in \{0.1, 0.5, 0.9\}$.

4.3. Change Scores

We employ the maximum likelihood (Eq. (2)) as the change score for the one-change-point functions, $\text{PT}_{\text{bias}}, \text{PT}_{\text{slope}}$, 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 computation of the change score in the multiple-input problem, the seafloor depth data, we use a spatial anomaly detection method AvgDiff (Kou et al., 2006). It computes the change score of a data point (x, y) by comparing the output y with the outputs of its spatial nearest neighbors $NN(x)$. The score is defined as $s(x, y) = \sum_{(x_i,y_i)\in NN(x)} w(x, x_i)\,|y - y_i|$,where $w(x, x_i)$ is a spatial weight between x and x_i. We use the inverse Euclidean distance as $w(x, x_i)$ and the 10-nearest neighbors as $NN(x)$.

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

4.4. Change-Point Estimation and Evaluation

For one-change-point functions, we assume there exists one change-point and estimate it by Eq. (1). We evaluate the absolute error between the estimated change-point and the ground-truth change-point.

For the multiple-change-point function, MCP, we assume we are not given the number of change-points. We estimate the change-points $\{\hat{x}_i^{\text{cp}}\}_{i=1}^{\hat{n}}$ using a

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

segmentation-based approach (Fryzlewicz, 2014) with the Bayesian information criterion penalty (Yao, 1988): $\Omega(\{x_i^{\text{cp}}\}_{i=1}^{\hat{n}}) = \frac{\hat{n}}{2}\log N$. The error is measured by the Hausdorff distance $e(\{x_i^{\text{cp}}\}_{i=1}^{n}, \{\hat{x}_i^{\text{cp}}\}_{i=1}^{\hat{n}}) = \max\{\max_{x^{\text{cp}}\in\{x_i^{\text{cp}}\}_{i=1}^{n}} \min_{\hat{x}^{\text{cp}}\in\{\hat{x}_i^{\text{cp}}\}_{i=1}^{\hat{n}}} |x^{\text{cp}} - \hat{x}^{\text{cp}}|, \max_{\hat{x}^{\text{cp}}\in\{\hat{x}_i^{\text{cp}}\}_{i=1}^{\hat{n}}} \min_{x^{\text{cp}}\in\{x_i^{\text{cp}}\}_{i=1}^{n}} |x^{\text{cp}} - \hat{x}^{\text{cp}}|\}$ (Truong et al., 2018).

For the multiple-input function, the seafloor depth dataset, we may consider the inputs with rapid depth changes as change-points, and however there are no ground-truths. In this experiment, we use the change-points estimated using all the data as the ground-truth change-points. In particular, we compute the ground-truth change scores using all the 72,800 data points (Figure 8(b)). We define the inputs with the top k change scores and the others as the ground-truth change-points and non-change-points, respectively. For each method, we evaluate precision@k for change scores at the same inputs as the ground-truths, which are estimated by posterior mean of a Gaussian process. Because the number of change-points is assumed to be few compared to non-change-points, we set $k = \{728(1\%), 3640(5\%), 7280(10\%), 10920(15\%), 14560\ (20\%)\}$.

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 values at each iteration, and evaluate a mean value over the B iterations and a final value after the B iterations. We conduct each experiment for 30 times.

4.5. Setting of Proposed Method

The proposed method is a meta-algorithm, and it needs the specifications of the underlying Bayesian optimization method, in particular, a covariance function $k(x, x')$ and an acquisition function $a(x)$. We use the ARD Matérn 5/2 kernel (M52) (Snoek et al., 2012): $k(x, x') = \theta_0 \left(1 + \sqrt{5r^2(x, x')} + \frac{5}{3}r^2(x, x')\right) \exp\left(-\sqrt{5r^2(x, x')}\right)$, where $r^2(x, x') = \Sigma_{d=1}^{D}\left(x^{(d)} - x'^{(d)}\right)^2 / \theta^{(d)^2}$. To study to what extent we should place priority on the uncertainty and exploration, we compare two acquisition functions: the expected improvement (EI) (Mockus et al., 1978) and the upper confidence bound (UCB) (Srinivas et al., 2010). We set the hyperparameter of the UCB algorithm as $\kappa \in \{0, 3, 6, 9, \infty\}$, where $\kappa = 0$ and $\kappa = \infty$ corresponds to a full exploitation strategy and a full exploration strategy, respectively. The hyperparameters of the covariance function $\{\theta_0, \theta_1, \ldots, \theta_D, \lambda^2\}$ are optimized by the empirical Bayes method at every time when a new observation is obtained.

4.6. Results and Discussions

Table 1 and 2 show the results of the mean error over iterations and the final error after iterations, respectively, for the target functions except the seafloor depth data. For PT_{bias}, PT_{slope} and NI, the proposed meta-algorithm with the EI and the UCB ($\kappa = 3, 6, 9$) performed well. Especially, all the results of the EI except for the MCP mean error over iterations are better then the comparing methods. Through the iterations, the proposed methods effectively decreases the errors as shown in Figure 6. In contrast, the greedy approaches (the UCB ($\kappa = 0$) and the ϵ-greedy strategy) showed poor results. This might be because the greedy approaches fell into the local optimums. As discussed in Section 3, change scores may vary depending on data. When the computed change scores at the ground-truth non-change-points were incorrectly high, the greedy approaches might fall into incorrect solutions. Especially the full exploitation strategy without any exploration, the UCB ($\kappa = 0$) did not improve the errors well after falling into the incorrect solutions as shown in Figure 6. On the other hand, the proposed meta-algorithm not only exploits the knowledge, but also explores uniformly in the input space in the early stages, which could provide the high change scores on the ground-truth change-points. Thanks to the property, the proposed meta-algorithm could avoid falling into the incorrect solutions and demonstrated the performance.

For MCP, the greedy approaches (the UCB ($\kappa = 0$) and the ϵ-greedy strategy ($\epsilon = 0.1$)) also showed the worse results than the exploration approaches (Table 1 and 2). This might be because it is required to explore uniformly in the input domain for the distributed three change-points. The proposed meta-algorithm with the UCB ($\kappa \geq 6$), which place priority rather than exploration, demonstrated the performance equal to or slightly better than the comparing methods. This might be derived from the inherently difficult multiple change-point detection without knowing their number, for which the errors of each method were not significantly different. The proposed meta-algorithm with the UCB ($\kappa \geq 6$) exploited the knowledge a little while exploring, which might provide the slightly better performance than the comparing methods for the final errors after iterations.

For the seafloor depth data, Table 3 and 4 show the mean precision@k over iterations and the final precision@k after iterations, respectively. The proposed methods without a full exploitation UCB ($\kappa = 0$) performed better than the comparing methods for all the different k. Especially, the EI and the UCB ($\kappa = 9$) worked well for the mean precision@k over iterations and the final precision@k after iterations, respectively. This might 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 with the two-dimensional input. Figure 8(c,d,e,f) shows the observed data and the posterior mean by the Gaussian process of the proposed methods with the EI, the UCB ($\kappa = 9$), the ϵ-greedy ($\epsilon = 0.1$) search, and the random search. The 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 8(b)) computed using $72,800$ data points. In contrast, the ϵ-greedy ($\epsilon = 0.1$) search (Figure 8(e)) only explored the few change-points and not the others. The change scores by the random search (Figure 8(f)) do not correspond to the ground-truths clearly in the right bottom part.

5. Related Work

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

Change-point detection (Basseville and Nikiforov, 1993; Gustafsson, 2000) is the task of finding abrupt changes in time-series data. There are several closely related tasks such as concept drift (Gama et al., 2014), event detection (Guralnik and Srivastava, 1999), and time-series segmentation (Keogh et al., 2001; Fryzlewicz, 2014). Spatial outlier detection (Kou et al., 2006) is the task of detecting areas that are significantly different from the other areas in spatial data. Spatial outlier detection and image segmentation (Pal and Pal, 1993) can be

Table 1: Means and standard deviations of the mean errors over iterations for 30 trials.

Method	PT_{bias}	PT_{slope}	NI	MO	MCP
ACPD (EI)	**23.6 ± 20.1**	23.6 ± 8.8	0.145 ± 0.117	15.4 ± 13.6	33.8 ± 6.7
ACPD (UCB $\kappa = 0$)	37.7 ± 29.0	35.9 ± 25.9	0.760 ± 0.657	24.5 ± 26.9	60.6 ± 21.2
ACPD (UCB $\kappa = 3$)	27.5 ± 20.8	**21.5 ± 9.1**	**0.123 ± 0.082**	16.6 ± 18.0	32.5 ± 4.2
ACPD (UCB $\kappa = 6$)	28.9 ± 18.4	26.0 ± 12.2	0.138 ± 0.091	**15.3 ± 16.5**	30.4 ± 2.8
ACPD (UCB $\kappa = 9$)	27.9 ± 23.8	25.5 ± 12.6	0.173 ± 0.126	17.0 ± 13.4	29.5 ± 1.9
ACPD (UCB $\kappa = \infty$)	45.2 ± 33.0	32.4 ± 11.6	0.242 ± 0.180	21.4 ± 8.8	**28.4 ± 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

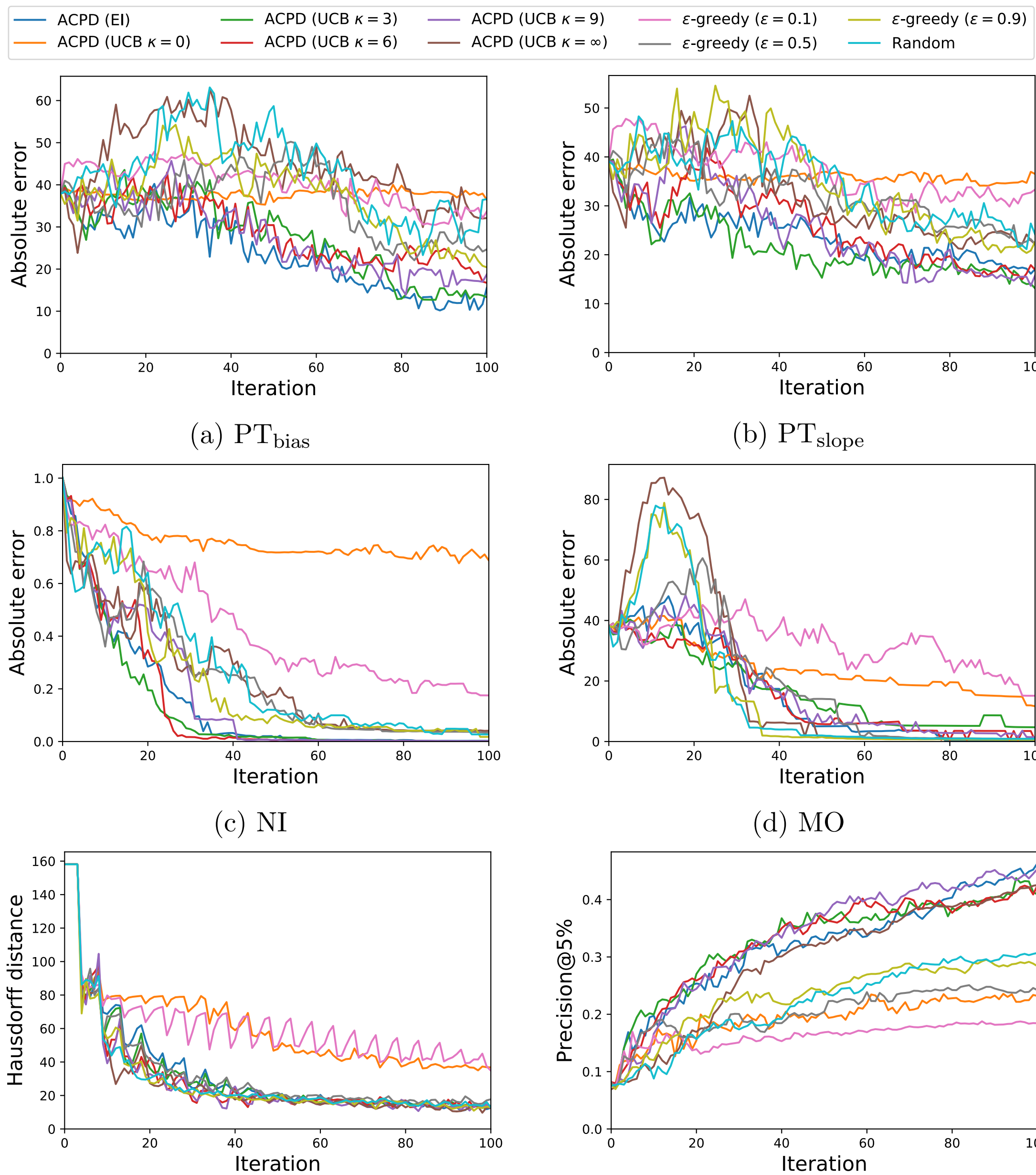
Table 2: Means and standard deviations of the final errors after iterations for 30 trials.

Method	PT_{bias}	PT_{slope}	NI	MO	MCP
ACPD (EI)	15.5 ± 34.9	17.0 ± 18.3	0.0026 ± 0.0030	**0.1 ± 0.1**	12.6 ± 7.3
ACPD (UCB $\kappa = 0$)	36.4 ± 29.5	36.5 ± 26.5	0.6875 ± 0.6777	11.7 ± 26.3	36.8 ± 27.6
ACPD (UCB $\kappa = 3$)	**13.3 ± 24.2**	**13.1 ± 12.0**	**0.0008 ± 0.0009**	4.7 ± 18.0	13.5 ± 7.0
ACPD (UCB $\kappa = 6$)	16.7 ± 31.2	16.9 ± 14.6	0.0011 ± 0.0018	0.3 ± 1.4	**12.5 ± 8.6**
ACPD (UCB $\kappa = 9$)	18.7 ± 34.5	15.8 ± 12.2	0.0014 ± 0.0015	1.4 ± 6.1	12.8 ± 5.6
ACPD (UCB $\kappa = \infty$)	32.1 ± 45.7	22.2 ± 15.5	0.0410 ± 0.0596	0.9 ± 0.8	**12.5 ± 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

also considered as multiple-input change-point detection problems. In general, change-point detection assumes 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 of the settings, the costs of acquiring data is not usually considered, and their goals are just to give a label of 'change' or 'normal' to each data point. On the other hand, in the ACPD problem, the target data is not usually a time series data, as in (Gijbels et al., 1999), and the data is acquired in an active manner by paying some cost of measurement.

ACPD is also related to experimental design (Chaloner and Verdinelli, 1995) where statistical models are used to determine efficient and effective series of experimental decisions. Bayesian optimization (Mockus et al., 1978; Snoek et al., 2012) suggests the next input to explore based on past observations to seek out the optimum of a black-box function that is expensive to evaluate. Gaussian processes (Rasmussen and Williams, 2006) are the typical choice as a model for the black-box function because it can handle uncertainty in the model. Various acquisition function are used to determine the next input by balancing exploration and exploitation based on the posterior mean and variance of the Gaussian processes. We use the Bayesian optimization technique as the key component of our proposed method, because our objective is also to find the optimum in a black-box change score function.

Active learning (Settles, 2012) 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 output a classifier that minimizes the predictive classification



(a) PT_{bias} (b) PT_{slope}

(c) NI (d) MO

(e) MCP (f) Seafloor depth

Figure 6: Error curves and a precision@5% curve. The proposed method efficiently decreases the error faster than the comparing methods for (a) PT_{bias}, (b) PT_{slope}, (c)NI, and (d) seafloor depth.

loss by choosing informative data points. The idea of active learning is applied to anomaly detection (Das et al., 2016), where a data point is queried to be labeled anomaly or normal.

The most related work to ACPD is the interactive image segmentation (Vezhnevets et al., 2012). In the setting, all the pixel values of an image are given, and, at each iteration, a set of pixels is queried an oracle for its label to segment the image. The problem may be

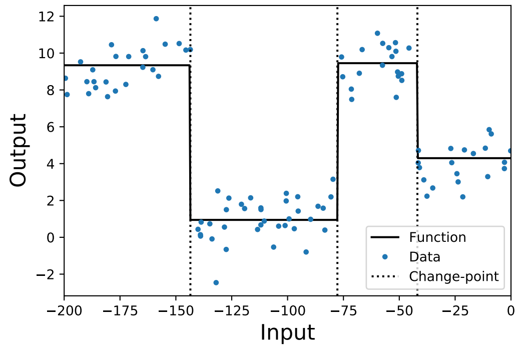
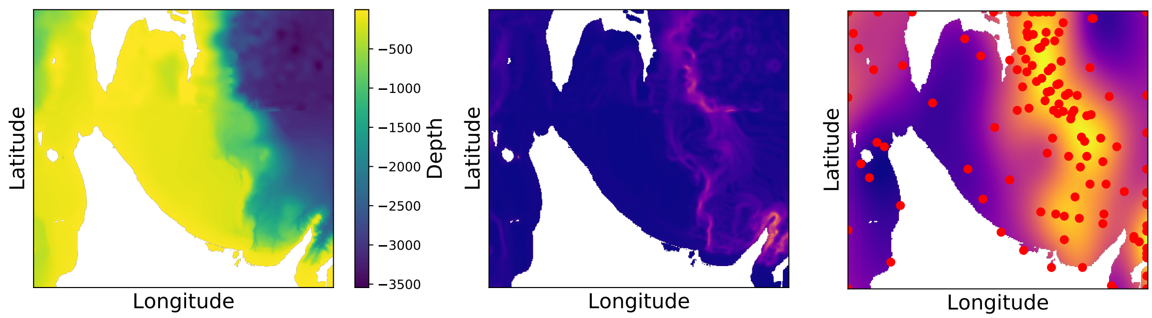
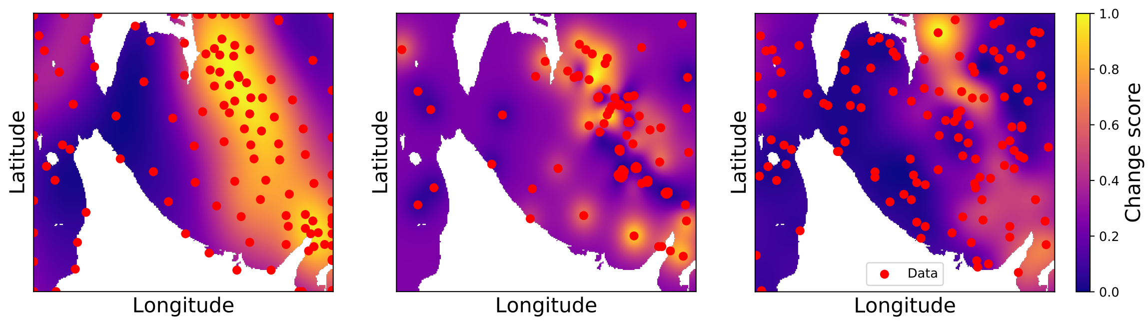


Figure 7: Piecewise-constant multi-change-point function (MCP), where change-points are located at −143.5, −77.7 and −41.9, and its 100 noisy observations.



(a) Seafloor depth data (b) Ground-truth (c) ACPD (EI)



(d) ACPD (UCB $\kappa = 9$) (e) ϵ-greedy ($\epsilon = 0.1$) (f) Random

Figure 8: (a) The 280 × 260 seafloor depth data. (b) The ground-truth change scores. (c),(d),(e),(f) 120 data points and change scores estimated by posterior mean of a Gaussian process, using the proposed methods (ACPD) with the EI and the UCB ($\kappa = 9$), the ϵ-greedy search (ϵ=0.1), and the random search. The proposed methods explored the ground-truth change-scores and non-change-points in a balanced manner, which produced the scores correspond to the ground-truths well using a small number of data.

regarded as an ACPD problem, where change-points are the boundaries of the segmentation, input is a two-dimensional coordinate of an image, and output is a value and label of a pixel; however, it is special that all the pixel values as a part of the output are given. Another

Table 3: Means and standard deviations of mean precision@k of each method over iterations for 30 trials, where k is indicated by percentage of the number of data.

Method	Precision@1%	Precision@5%	Precision@10%	Precision@15%	Precision@20%
ACPD (EI)	**0.145 ± 0.072**	0.321 ± 0.057	0.446 ± 0.050	0.497 ± 0.034	0.543 ± 0.027
ACPD (UCB $\kappa = 0$)	0.056 ± 0.054	0.193 ± 0.049	0.291 ± 0.052	0.347 ± 0.048	0.398 ± 0.052
ACPD (UCB $\kappa = 3$)	0.137 ± 0.051	0.331 ± 0.037	0.456 ± 0.038	0.502 ± 0.032	0.546 ± 0.026
ACPD (UCB $\kappa = 6$)	0.116 ± 0.061	0.328 ± 0.036	0.464 ± 0.032	0.510 ± 0.028	0.557 ± 0.027
ACPD (UCB $\kappa = 9$)	0.106 ± 0.050	**0.337 ± 0.037**	**0.472 ± 0.037**	**0.515 ± 0.035**	**0.561 ± 0.031**
ACPD (UCB $\kappa = \infty$)	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: Means and standard deviations of precision@k of each method after iterations for 30 trials, where k is indicated by percentage of the number of data.

Method	Precision@1%	Precision@5%	Precision@10%	Precision@15%	Precision@20%
ACPD (EI)	**0.269 ± 0.175**	**0.464 ± 0.062**	**0.584 ± 0.052**	**0.613 ± 0.041**	0.638 ± 0.031
ACPD (UCB $\kappa = 0$)	0.084 ± 0.119	0.232 ± 0.077	0.324 ± 0.083	0.368 ± 0.089	0.393 ± 0.090
ACPD (UCB $\kappa = 3$)	0.260 ± 0.113	0.409 ± 0.072	0.542 ± 0.065	0.587 ± 0.051	0.620 ± 0.041
ACPD (UCB $\kappa = 6$)	0.207 ± 0.159	0.425 ± 0.081	0.553 ± 0.056	0.593 ± 0.042	0.632 ± 0.030
ACPD (UCB $\kappa = 9$)	0.192 ± 0.117	0.454 ± 0.059	0.580 ± 0.054	0.607 ± 0.037	0.637 ± 0.030
ACPD (UCB $\kappa = \infty$)	0.074 ± 0.091	0.426 ± 0.055	0.575 ± 0.048	0.593 ± 0.033	**0.640 ± 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

related work is the active learning with drifting streaming data (Žliobaitė et al., 2014). At each iteration, we obtain a piece of data, and we determine if we query an oracle for its class label to train a classifier for drifting streaming data. This problem is also similar to ACPD, when we see input is time and output is the data and its label; however, it is different that the data as a part of the output is given and that the input is the time.

6. Conclusion

We proposed ACPD, a novel active learning problem for cost-efficient change-point detection in a black-box expensive-to-evaluate function. In ACPD, we determine the next input to be explored in order to find the change-point in the function with as few evaluations as possible. We also proposed a general framework that does not depend on the types of data or change-points, by reusing an existing change-point detection method and a Bayesian optimization technique. The experimental results showed that the proposed meta-algorithm outperformed the comparing methods in consideration of exploration and exploitation.

Acknowledgments

This research was supported by JSPS KAKENHI Grant Number 15H01704, 18J12982 and 18H03287.

References

Mikhail Alekseevich Anisimov. *Critical phenomena in liquids and liquid crystals.* Gordon and Breach Science Publishers, 1991.

Michèle Basseville and Igor V. Nikiforov. *Detection of abrupt changes: theory and application.* Prentice Hall, 1993.

Kathryn Chaloner and Isabella Verdinelli. Bayesian experimental design: A review. *Statistical Science*, 10(3):273–304, 1995.

Shubhomoy Das, Weng-Keen Wong, Thomas Dietterich, Alan Fern, and Andrew Emmott. Incorporating expert feedback into active anomaly discovery. In *Proceedings of the 16th IEEE International Conference on Data Mining (ICDM)*, pages 853–858, 2016.

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

João Gama, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. A survey on concept drift adaptation. *ACM Computing Surveys*, 46(4): 44:1–44:37, 2014.

Irene Gijbels, Peter Hall, and Aloïs Kneip. On the estimation of jump points in smooth curves. *Annals of the Institute of Statistical Mathematics*, 51(2):231–251, 1999.

Valery Guralnik and Jaideep Srivastava. Event detection from time series data. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 33–42, 1999.

Fredrik Gustafsson. *Adaptive filtering and change detection.* Wiley, 2000.

Tsuyoshi Idé, Dzung T. Phan, and Jayant Kalagnanam. Change detection using directional statistics. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1613–1619, 2016.

Yoshinobu Kawahara, Takehisa Yairi, and Kazuo Machida. Change-point detection in time-series data based on subspace identification. In *Proceedings of the Seventh IEEE International Conference on Data Mining (ICDM)*, pages 559–564, 2007.

Eamonn J. Keogh, Selina Chu, David M. Hart, and Michael J. Pazzani. An online algorithm for segmenting time series. In *Proceedings of the First IEEE International Conference on Data Mining (ICDM)*, pages 289–296, 2001.

Yufeng Kou, Chang-Tien Lu, and Dechang Chen. Spatial weighted outlier detection. In *Proceedings of the 2006 SIAM International Conference on Data Mining (SDM)*, pages 614–618, 2006.

Song Liu, Makoto Yamada, Nigel Collier, and Masashi Sugiyama. Change-point detection in time-series data by relative density-ratio estimation. *Neural Networks*, 43:72–83, 2013.

Jonas Mockus, Vytautas Tiesis, and Antanas Zilinskas. *Toward Global Optimization*, volume 2, chapter The application of Bayesian Methods for Seeking the Extremum, pages 117–129. Elsevier, 1978.

Nikhil R. Pal and Sankar K. Pal. A review on image segmentation techniques. *Pattern Recognition*, 26(9):1277 – 1294, 1993.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian processes for machine learning*. MIT Press, 2006.

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.

Burr Settles. *Active learning*. Morgan & Claypool Publishers, 2012.

Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems (NIPS) 25*, pages 2951–2959, 2012.

Niranjan Srinivas, Andreas Krause, Sham Kakade, and Matthias W. Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proceedings of the 27th International Conference on Machine Learning (ICML)*, pages 1015–1022, 2010.

Charles Truong, Laurent Oudre, and Nicolas Vayatis. A review of change point detection methods. *CoRR*, abs/1801.00718, 2018. URL http://arxiv.org/abs/1801.00718.

Alexander Vezhnevets, Joachim M. Buhmann, and Vittorio Ferrari. Active learning for semantic segmentation with expected change. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3162–3169, 2012.

Indrė Žliobaitė, Albert Bifet, Bernhard Pfahringer, and Geoffrey Holmes. Active learning with drifting streaming data. *IEEE Transactions on Neural Networks and Learning Systems*, 25(1):27–39, 2014.

Yu Wang, Aniket Chakrabarti, David Sivakoff, and Srinivasan Parthasarathy. Fast change point detection on dynamic social networks. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2992–2998, 2017.

Kenji Yamanishi, Jun'ichi Takeuchi, Graham J. Williams, and Peter Milne. On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms. *Data Mining and Knowledge Discovery*, 8(3):275–300, 2004.

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