Sequential Change-Point Detection Based on Direct Density-Ratio Estimation

Yoshinobu Kawahara
Osaka University
kawahara@ar.sanken.osaka-u.ac.jp

Masashi Sugiyama
Tokyo Institute of Technology
sugi@cs.titech.ac.jp

Abstract

Change-point detection is the problem of discovering time points at which properties of time-series data change. This covers a broad range of real-world problems and has been actively discussed in the community of statistics and data mining. In this paper, we present a novel non-parametric approach to detecting the change of probability distributions of sequence data. Our key idea is to estimate the ratio of probability densities, not the probability densities themselves. This formulation allows us to avoid non-parametric density estimation, which is known to be a difficult problem. We provide a change-point detection algorithm based on direct density-ratio estimation that can be computed very efficiently in an online manner. The usefulness of the proposed method is demonstrated through experiments using artificial and real-world datasets.

1 Introduction

The problem of discovering time points at which properties of time-series data change is attracting a lot of attention in the data mining community [5, 7, 15, 24]. This problem is referred to as change-point detection [37, 18, 21] or event detection [3], and covers a broad range of real-world problems such as fraud detection in cellular systems [29, 6], intrusion detection in computer networks [38], irregular-motion detection in vision systems [22], signal segmentation in data stream [5], and fault detection in engineering systems [13].

The problem of change-point detection has been actively studied over the last several decades in statistics. A typical statistical formulation of change-point detection is to consider probability distributions from which data in the past and present intervals are generated, and regard the target time point as a change point if two distributions are significantly different. Various approaches to change-point detection have been investigated within this statistical framework, including the CUSUM (cumulative sum) [5] and GLR (generalized likelihood ratio) [14, 15] approaches. In these approaches, *the logarithm of the likelihood ratio* between two consecutive invervals in time-series data is monitored for detecting change points. In recent years, the above statistical framework has been extensively explored in the data mining community in connection with real-world applications, for example, approaches based on novelty detection [26, 27], maximum-likelihood estimation [3], and online learning of autoregressive models [37, 36]. Another line of research is based on analysis of subspaces in which time-series sequences are constrained [28, 18, 21]. This approach has a strong connection with a system identification method called *subspace identification*, which has been thoroughly studied in the area of control theory [31, 20]. Moreover, there are several literatures on spectrum-based methods for change-point detection, where the changes in autocorrelation structure are caught through the Fourier or wavelet-based spectrum analysis [1, 30, 8]. Also, we should mention that in the offline case, once a class of models for change-point detection is fixed, the problem becomes to search a combination of time points that give a sequence of the models fitting into data, which has been studied actively based on several combinatorial techniques [4, 10].

A common limitation of the above-mentioned approaches is that they rely on pre-specified parametric models such as probability density models, autoregressive models and state-space models, or some specific quantities such as averages, covariances, autocorrelation and spectrums. Thus, these methods tend to be less flexible in real-world change-point detection scenarios. The primal purpose of this paper is to present a more flexible non-parametric method that does not rely on a strong model assumption. In the community of statistics, some non-parametric approaches to change-point detection problems have been explored, in which non-parametric density estimation is used for calculating the likelihood ratio [9, 7]. However, since non-parametric density estimation is known to be a hard problem [16, 17], this naive approach may not be promising in practice.

Our key idea for alleviating this difficulty is to directly estimate the *ratio* of probability densities, not the probability densities themseleves. Namely, we estimate the density ratio (which is also referred to as the *importance* in literature [12]) without going through density estimation. Recently, direct density-ratio estimation has been actively explored in the machine learning community, e.g., Kernel Mean Matching [17] and the Kullback-Leibler Importance Estimation Procedure (KLIEP) [34, 19]. The KLIEP is shown to give the optimal convergence rate (minmax rate) for non-parametric density-ratio estimation [34]. However, these conventional methods are batch algorithms and are not suitable for change-point detection due to the sequential nature of time-series data analysis. To cope with this problem, we give an online version of the KLIEP algorithm and develop a flexible and computationally efficient change-point detection method. An advantage of our method over existing non-parametric approaches such as the sequential one-class support vector machine [32, 11] is that our method is equipped with a natural cross validation procedure. Thus, the values of tuning parameters such as the kernel bandwidth can be objectively detemined in our method. This is a highly useful property in unsupervised change detection scenarios.

The remainder of this paper is organized as follows. In Section 2, we formulate the change-point detection problem of time-series data as a density-ratio estimation problem. In Section 3, we develop a new change-point detection algorithm based on an online extension of a direct density-ratio estimation procedure. Finally, we report experimental results on artificial and real-world datasets in Section 4 and conclude by summarizing our contribution and possible future works in Section 5.

2 Problem Formulation and Basic Approach

In this section, we formulate a change-point detection problem based on the density ratio in Section 2.1 and describe our basic approach to this problem in Section 2.2.

2.1 Problem Formulation

Let $\boldsymbol{y}(t)$ ($\in \mathbb{R}^d$) be a d-dimensional time series sample at time t. Our task is to detect whether there exists a change point between two consecutive time intervals, called the *reference* and *test* intervals. The conventional algorithms [5, 15] consider the likelihood ratio over samples from the two intervals. However, time-series samples are generally not independent over time and therefore directly dealing with non-independent samples is cumbersome. To ease this difficulty, in this paper, we consider *sequences* of samples in each time intervals; let $\boldsymbol{Y}(t)$ ($\in \mathbb{R}^{dk}$) be the forward subsequence of length k at time t:

$$\boldsymbol{Y}(t) = \left[\boldsymbol{y}(t)^T, \boldsymbol{y}(t+1)^T, \ldots, \boldsymbol{y}(t+k-1)^T\right]^T,$$

where $\bullet^T$ denotes the transpose of a vector or matrix. This is a common practice in subspace identification since it allows us to implicitly take time correlation into consideration to some degree [31, 20]. Our algorithm stated in the remainder of this paper is based on the logarithm of the likelihood ratio of the *sequence sample* $\boldsymbol{Y}$ defined by

$$s(\boldsymbol{Y}) = \ln \frac{p_{\mathrm{te}}(\boldsymbol{Y})}{p_{\mathrm{rf}}(\boldsymbol{Y})},$$

where $p_{\mathrm{te}}(\boldsymbol{Y})$ and $p_{\mathrm{rf}}(\boldsymbol{Y})$ are the probability density functions of the reference and test sequence samples, respectively.

Let t_{rf} and t_{te} ($t_{\mathrm{rf}} < t_{\mathrm{te}}$) be the starting time points of the reference and test intervals, respectively. Suppose we have n_{rf} and n_{te} sequence samples whose starting points are in the reference and test intervals, respectively. Then,

$$t_{\mathrm{te}} = t_{\mathrm{rf}} + n_{\mathrm{rf}}.$$

The above formulation is summarized in Figure 1.

For brevity, we use the following shorthand notation in the sequel:

$$\begin{aligned}\boldsymbol{Y}_{\mathrm{rf}}(i) &= \boldsymbol{Y}(t_{\mathrm{rf}} + i - 1),\\ \boldsymbol{Y}_{\mathrm{te}}(i) &= \boldsymbol{Y}(t_{\mathrm{te}} + i - 1).\end{aligned}$$

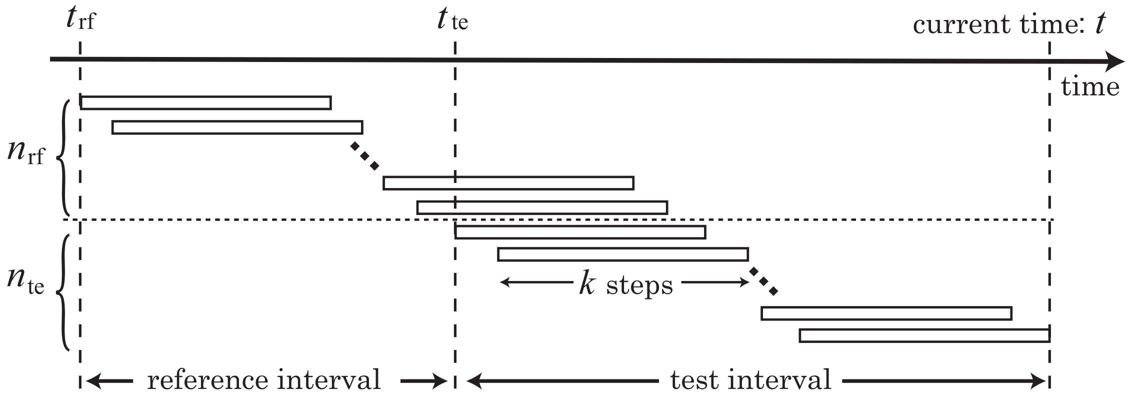


Figure 1: Definition of the reference and test intervals.

Thus, the i-th reference and test samples are denoted by $\mathbf{Y}_{\mathrm{rf}}(i)$ and $\mathbf{Y}_{\mathrm{te}}(i)$, respectively. Now, let us introduce the following hypotheses about the observations[1]:

$$\begin{aligned} H_0: \quad & p(\boldsymbol{Y}(i)) = p_{\mathrm{rf}}(\boldsymbol{Y}(i)) \quad \text{for } t_{\mathrm{rf}} \leq i < t. \\ H_1: \quad & p(\boldsymbol{Y}(i)) = p_{\mathrm{rf}}(\boldsymbol{Y}(i)) \quad \text{for } t_{\mathrm{rf}} \leq i < t_{\mathrm{te}}, \\ & p(\boldsymbol{Y}(i)) = p_{\mathrm{te}}(\boldsymbol{Y}(i)) \quad \text{for } t_{\mathrm{te}} \leq i < t. \end{aligned}$$

Then the likelihood ratio between the hypotheses H_0 and H_1 is

$$\begin{aligned} \Lambda &= \frac{\prod_{i=1}^{n_{\mathrm{rf}}} p_{\mathrm{rf}}(\mathbf{Y}_{\mathrm{rf}}(i)) \prod_{i=1}^{n_{\mathrm{te}}} p_{\mathrm{te}}(\mathbf{Y}_{\mathrm{te}}(i))}{\prod_{i=1}^{n_{\mathrm{rf}}} p_{\mathrm{rf}}(\mathbf{Y}_{\mathrm{rf}}(i)) \prod_{i=1}^{n_{\mathrm{te}}} p_{\mathrm{rf}}(\mathbf{Y}_{\mathrm{te}}(i))} \\ &= \frac{\prod_{i=1}^{n_{\mathrm{te}}} p_{\mathrm{te}}(\mathbf{Y}_{\mathrm{te}}(i))}{\prod_{i=1}^{n_{\mathrm{te}}} p_{\mathrm{rf}}(\mathbf{Y}_{\mathrm{te}}(i))}. \end{aligned}$$

Thus, we can decide whether there is a change point between the reference and test intervals by monitoring the logarithm of the likelihood ratio:

$$S = \sum_{i=1}^{n_{\mathrm{te}}} \ln \frac{p_{\mathrm{te}}(\mathbf{Y}_{\mathrm{te}}(i))}{p_{\mathrm{rf}}(\mathbf{Y}_{\mathrm{te}}(i))}. \tag{1}$$

Based on the logarithm of the likelihood ratio S, we conclude

$$\begin{cases} S \leq \mu & \longrightarrow \quad \text{no change occurs,} \\ \text{otherwise} & \longrightarrow \quad \text{a change occurs,} \end{cases} \tag{2}$$

where μ (> 0) is a predetermined threshold for the decision of a change point.

[1] This formulation is slightly different from the original one used in the CUSUM or GLR algorithms, where the tested time point is not fixed at time t_{te}.

The remaining question of this procedure is how to calculate the density ratio,

$$w(\boldsymbol{Y}) := \frac{p_{\mathrm{te}}(\boldsymbol{Y})}{p_{\mathrm{rf}}(\boldsymbol{Y})},$$

because, in practice, the above ratio is unknown and therefore we need to estimate it from samples. A naive approach to this would be to first estimate the reference and test densities separately from the reference and test sequence samples, and then estimate the density ratio by taking the ratio of the estimated densities. However, since non-parametric density estimation is known to be a hard problem [16, 17], this naive approach to change-point detection via non-parametric density estimation may not be effective—directly estimating the density ratio without estimating the densities would be more promising.

2.2 Direct Density-Ratio Estimation

As described above, we need to estimate the density ratio for solving the change-point detection problem. Here, we show how the density ratio could be directly estimated without going through density estimation based on the Kullback-Leibler Importance Estimation Procedure (KLIEP) [34].

Let us model the density ratio $w(\boldsymbol{Y})$ by a non-parametric Gaussian kernel model:

$$\widehat{w}(\boldsymbol{Y}) = \sum_{l=1}^{n_{\mathrm{te}}} \alpha_l K_\sigma(\boldsymbol{Y}, \boldsymbol{Y}_{\mathrm{te}}(l)), \tag{3}$$

where $\{\alpha_l\}_{l=1}^{n_{\mathrm{te}}}$ are parameters to be learned from data samples and $K_\sigma(\boldsymbol{Y}, \boldsymbol{Y}')$ is the Gaussian kernel function with mean $\boldsymbol{Y}'$ and standard deviation σ:

$$K_\sigma(\boldsymbol{Y}, \boldsymbol{Y}') = \exp\left(-\frac{\|\boldsymbol{Y} - \boldsymbol{Y}'\|^2}{2\sigma^2}\right). \tag{4}$$

Using the model $\widehat{w}(\boldsymbol{Y})$, we can estimate the test interval density $p_{\mathrm{te}}(\boldsymbol{Y})$ by

$$\widehat{p}_{\mathrm{te}}(\boldsymbol{Y}) = \widehat{w}(\boldsymbol{Y}) p_{\mathrm{tr}}(\boldsymbol{Y}).$$

The parameters $\{\alpha_l\}_{l=1}^{n_{\mathrm{te}}}$ in the model (3) are determined so that the empirical Kullback-Leibler divergence from $p_{\mathrm{te}}(\boldsymbol{Y})$ to $\widehat{p}_{\mathrm{te}}(\boldsymbol{Y})$ is minimized. The solution of this problem can be obtained by solving the following convex optimization problem:

$$\begin{cases} \displaystyle\max_{\{\alpha_l\}_{l=1}^{n_{\mathrm{te}}}} \sum_{i=1}^{n_{\mathrm{te}}} \log\left(\sum_{l=1}^{n_{\mathrm{te}}} \alpha_l K_\sigma(\boldsymbol{Y}_{\mathrm{te}}(i), \boldsymbol{Y}_{\mathrm{te}}(l))\right), \\ \displaystyle\text{s.t. } \frac{1}{n_{\mathrm{rf}}} \sum_{i=1}^{n_{\mathrm{rf}}} \sum_{l=1}^{n_{\mathrm{te}}} \alpha_l K_\sigma(\boldsymbol{Y}_{\mathrm{rf}}(i), \boldsymbol{Y}_{\mathrm{te}}(l)) = 1, \\ \quad \text{and} \quad \alpha_1, \ldots, \alpha_{n_{\mathrm{te}}} \geq 1. \end{cases} \tag{5}$$

The equality constraint in the above optimization problem comes from the requirement that $\widehat{w}(\boldsymbol{Y})$ should be properly normalized since $p_{\mathrm{rf}}(\boldsymbol{Y})\widehat{w}(\boldsymbol{Y})$ $(= \widehat{p}_{\mathrm{te}}(\boldsymbol{Y}))$ is a probability density

```
   input: Reference samples $\mathcal{Y}_{\mathrm{rf}} = \{\boldsymbol{Y}_{\mathrm{rf}}(i)\}_{i=1}^{n_{\mathrm{rf}}}$, test samples $\mathcal{Y}_{\mathrm{te}} = \{\boldsymbol{Y}_{\mathrm{te}}(i)\}_{i=1}^{n_{\mathrm{te}}}$, and the
          Gaussian width $\sigma$.
1  $\boldsymbol{K}_{i,l} = K_\sigma(\boldsymbol{Y}_{\mathrm{te}}(i), \boldsymbol{Y}_{\mathrm{te}}(l))$ $(i = 1, \ldots, n_{\mathrm{te}}, l = 1, \ldots, n_{\mathrm{te}})$.
2  $\boldsymbol{b}_l = \frac{1}{n_{\mathrm{rf}}} \sum_{i=1}^{n_{\mathrm{rf}}} K_\sigma(\boldsymbol{Y}_{\mathrm{rf}}(i), \boldsymbol{Y}_{\mathrm{te}}(l))$ $(l = 1, \ldots, n_{\mathrm{te}})$.
3  Initialize $\boldsymbol{\alpha}$ $(> \boldsymbol{0})$ and $\epsilon$ $(0 < \epsilon \ll 1)$.
4  Repeat
5      Perform gradient ascent:
           $\boldsymbol{\alpha} \leftarrow \boldsymbol{\alpha} + \epsilon \boldsymbol{K}(\boldsymbol{1}./\boldsymbol{K}\boldsymbol{\alpha})$.
6      Perform feasibility satisfaction:
           $\boldsymbol{\alpha} \leftarrow \boldsymbol{\alpha} + (1 - \boldsymbol{b}^T\boldsymbol{\alpha})\boldsymbol{b}/(\boldsymbol{b}^T\boldsymbol{b})$,
           $\boldsymbol{\alpha} \leftarrow \max(\boldsymbol{0}, \boldsymbol{\alpha})$,
           $\boldsymbol{\alpha} \leftarrow \boldsymbol{\alpha}/(\boldsymbol{b}^T\boldsymbol{\alpha})$.
7  Until convergence;
8  Output $\widehat{w}(\boldsymbol{Y}) \leftarrow \sum_{l=1}^{n_{\mathrm{te}}} \boldsymbol{\alpha}_l K_\sigma(\boldsymbol{Y}, \boldsymbol{Y}_{\mathrm{te}}(l))$.
```

Algorithm 1: The KLIEP algorithm in pseudo code. '$./$' in Line 5 indicates the element-wise division. An inequality for a vector in Line 3 and the max operation for a vector in Line 6 are applied in the elment-wise manner.

function. The non-negativity constraint comes from the non-negativity of the density ratio function.

A pseudo code of the KLIEP is described in Algorithm 1, which will be used for initialization purposes in the online setup discussed in the next section. In the Gaussian kernel model (3), we set the Gaussian centers at the test subsequences $\{\boldsymbol{Y}_{\mathrm{te}}(i)\}_{i=1}^{n_{\mathrm{te}}}$, not the reference sequences. The reason for this choice is as follows. The density-ratio $w(\boldsymbol{Y})$ tends to take large values if the reference sequence density $p_{\mathrm{rf}}(\boldsymbol{Y})$ is small and the test sequence density $p_{\mathrm{te}}(\boldsymbol{Y})$ is large by definition. When a non-negative function is approximated by a Gaussian kernel model in general, many kernels may be needed in the region where the output of the target function is large. Thus, it would be effective to allocate many kernels at high reference density regions, which can be achieved by setting the centers at the test sequence points $\{\boldsymbol{Y}_{\mathrm{te}}(i)\}_{i=1}^{n_{\mathrm{te}}}$.

In Algorithm 1, the kernel width σ is an open tuning parameter and needs to be chosen appropriately for better estimation. As shown in the paper [34], the kernel width σ can be determined from data samples using *likelihood cross validation.* A pseudo code of likelihood cross validation is described in Algorithm 2.

3 Online Algorithm

The above framework would be a suitable method for flexible change-point detection since it enables us to directly compute a non-parametric estimate of the density-ratio $w(\boldsymbol{Y})$ without going through density estimation. However, it is a batch algorithm (i.e., all samples are

```
input: Reference samples $\mathcal{Y}_{\mathrm{rf}} = \{\boldsymbol{Y}_{\mathrm{rf}}(i)\}_{i=1}^{n_{\mathrm{rf}}}$, test samples $\mathcal{Y}_{\mathrm{te}} = \{\boldsymbol{Y}_{\mathrm{te}}(i)\}_{i=1}^{n_{\mathrm{te}}}$, and the
       Gaussian width candidates $\{\sigma_i\}_{i=1}^{s}$.
Split $\mathcal{Y}_{\mathrm{te}}$ into $R$ disjoint subsets $\{\mathcal{Y}_r\}_{r=1}^{R}$.
For each Gaussian width $\sigma_i$
    For each split $r = 1, \dots, R$
        Call KLIEP with model $\sigma_i$ and split $r$:
            $\widehat{w}_r(\boldsymbol{Y}) \leftarrow \mathrm{KLIEP}(\mathcal{Y}_{\mathrm{rf}}, \{\mathcal{Y}_j\}_{j \neq r}, \sigma_i)$.
        Compute out-of-sample log-likelihood:
            $\widehat{J}_r(i) \leftarrow \frac{1}{|\mathcal{Y}_r|} \sum_{\boldsymbol{Y} \in \mathcal{Y}_r} \log \widehat{w}_r(\boldsymbol{Y})$.
    Compute average out-of-sample log-likelihood:
        $\widehat{J}(i) \leftarrow \frac{1}{R} \sum_{r=1}^{R} \widehat{J}_r(i)$.
Choose the optimal Gaussian width:
    $\widehat{i} \leftarrow \arg\max_i \widehat{J}(i)$
Output $\widehat{w}(\boldsymbol{Y}) \leftarrow \mathrm{KLIEP}(\mathcal{Y}_{\mathrm{rf}}, \mathcal{Y}_{\mathrm{te}}, \sigma_{\widehat{i}})$.
```

Algorithm 2: Pseudo code of kernel width selection in KLIEP by likelihood cross validation.

used for estimation) and therefore is not computationally efficient in the current setup since solutions need to be computed sequentially over time in change-point detection. To reduce the computational cost, we develop an online version of the above algorithm that recursively builds a new solution upon the previous solution (Section 3.1). We then describe the overall procedure of our proposed algorithm for change-point detection in Section 3.2.

3.1 Online Density-Ratio Estimation

Let $\widehat{w}(\boldsymbol{Y})$ be an estimate of the density ratio by the offline algorithm (Algorithm 1) from $\{\boldsymbol{Y}_{\mathrm{rf}}(i)\}_{i=1}^{n_{\mathrm{rf}}}$ and $\{\boldsymbol{Y}_{\mathrm{te}}(i)\}_{i=1}^{n_{\mathrm{te}}}$ and let $\{\widehat{\alpha}_l\}_{l=1}^{n_{\mathrm{te}}}$ be the learned parameters:

$$\widehat{w}(\boldsymbol{Y}) = \sum_{l=1}^{n_{\mathrm{te}}} \widehat{\alpha}_l K_\sigma(\boldsymbol{Y}, \boldsymbol{Y}_{\mathrm{te}}(l)).$$

Let us consider the case where a new sample point $\boldsymbol{y}(t_{\mathrm{te}} + k + n_{\mathrm{te}})$ is observed—in the sequence representation, $\boldsymbol{Y}(t_{\mathrm{te}} + n_{\mathrm{te}} + 1)$ is additionally obtained (see Figure 2). Let $\widehat{w}'(\boldsymbol{Y})$ be an estimate of the density ratio by Algorithm 1 after the new sample is given. Here we give an online method that computes the new solution $\widehat{w}'(\boldsymbol{Y})$ efficiently from the previous solution $\widehat{w}(\boldsymbol{Y})$ and the new sample $\boldsymbol{Y}(t_{\mathrm{te}} + n_{\mathrm{te}} + 1)$.

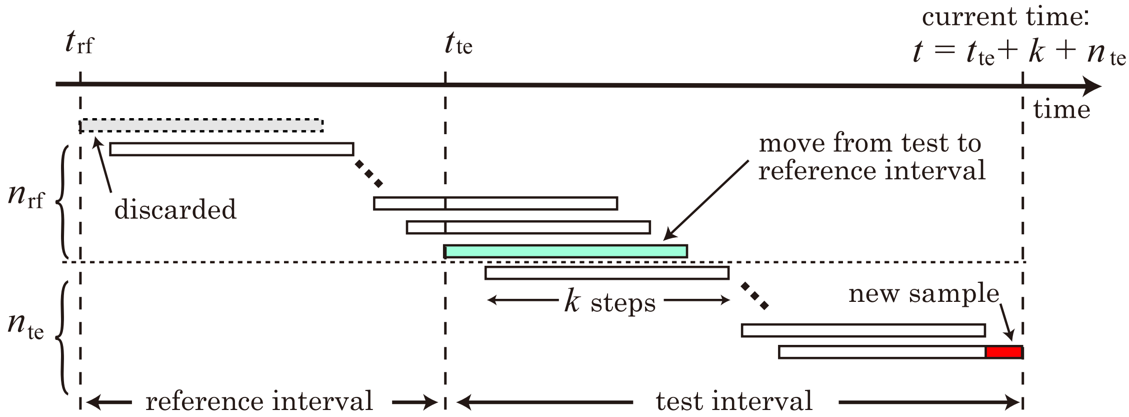


Figure 2: Online setup of change detection.

Our idea basically relies on *stochastic gradient descent* [2], which allows us to learn the parameters efficiently in an online manner. However, since the algorithm described in the previous section is a non-parametric kernel method and the basis functions vary over time, it is not straightforward to employ the techniques of stochastic gradient descent in the current setup. Recently, an online learning technique for kernel methods has been addressed [25]. Here, we apply this technique to the problem (5) and give an online version of the algorithm.

Let $E_t(w)$ be the empirical error for $\boldsymbol{Y}_{\mathrm{te}}(t)$:

$$E_i(w) = -\log w(\boldsymbol{Y}_{\mathrm{te}}(i)).$$

Note that the solution $\widehat{w}$ is given as the minimizer of $\sum_{i=1}^{n_{\mathrm{te}}} E_i(w)$ under the normalization and non-negativity constraints (cf. Eq.(5)). Let us assume that w is searched within a reproducing kernel Hilbert space $\mathcal{H}$. Then the following reproducing property holds:

$$\langle w(\cdot), K(\cdot, \boldsymbol{Y}')\rangle_{\mathcal{H}} = w(\boldsymbol{Y}'),$$

where $K(\cdot,\cdot)$ is the reproducing kernel of $\mathcal{H}$ and $\langle \cdot,\cdot\rangle_{\mathcal{H}}$ denotes the inner product in $\mathcal{H}$. Let us consider the following regularized empirical error:

$$E_i'(w) = -\log w(\boldsymbol{Y}_{\mathrm{te}}(i)) + \frac{\lambda}{2}\|w\|_{\mathcal{H}}^2,$$

where $\lambda(> 0)$ is the regularization parameter and $\|\cdot\|_{\mathcal{H}}$ denotes the norm in $\mathcal{H}$.

The basic idea of online learning for kernel methods is to update the solution as

$$\widehat{w}' \leftarrow \widehat{w} - \eta \partial_w E_{i+1}'(\widehat{w}),$$

where ∂_w denotes the partial derivative with respect to w and $\eta(> 0)$ is the learning rate that controls the adaptation sensitivity to the new sample. Given the Gaussian kernel model (3), the above update rule is explicitly expressed as

$$\widehat{w}' \leftarrow \widehat{w} - \eta \left(-\frac{K_\sigma(\cdot, \boldsymbol{Y}_{\mathrm{te}}(n_{\mathrm{te}}+1)}{\widehat{w}(\boldsymbol{Y}_{\mathrm{te}}(n_{\mathrm{te}}+1)} + \lambda \widehat{w} \right).$$

This implies that the parameters are updated as

$$\begin{cases} \widehat{\alpha}'_l \leftarrow (1-\eta\lambda)\widehat{\alpha}_{l+1} & \text{if } \ l = 1, \ldots, n_{\mathrm{te}}-1, \\ \widehat{\alpha}'_l \leftarrow \dfrac{\eta}{\widehat{w}(\boldsymbol{Y}_{\mathrm{te}}(n_{\mathrm{te}}+1)} & \text{if } \ l = n_{\mathrm{te}}. \end{cases}$$

Note that $\eta\lambda$, which should be chosen between zero and one, works as a *forgetting* factor for older samples. When the target time series is highly nonstationary, it is often useful to deemphasize the influence of older samples via the forgetting factor. If such weighting is not necessary, we merely set $\eta = 0$.

In addition to reducing the empirical error, the constraints in the optimizatin problem (5) need to be fullfilled. This could be achieved in the same manner as the batch algorithm (Algorithm 1), but normalization is carried out over $\{\boldsymbol{Y}_{\mathrm{rf}}(t+1)\}_{t=1}^{n_{\mathrm{rf}}}$ with Gaussian centers $\{\boldsymbol{Y}_{\mathrm{te}}(l+1)\}_{t=1}^{n_{\mathrm{te}}}$:

$$\frac{1}{n_{\mathrm{rf}}} \sum_{t=1}^{n_{\mathrm{rf}}} \sum_{l=1}^{n_{\mathrm{te}}} \alpha_l K_\sigma(\boldsymbol{Y}_{\mathrm{rf}}(t+1), \boldsymbol{Y}_{\mathrm{te}}(l+1)) = 1. \tag{6}$$

Altogether, the pseudo code of the online update procedure is summarized in Algorithm 3.

3.2 Change-Point Detection Algorithm

The entire procedure for change-point detection based on direct density-ratio estimation is summarized in Algorithm 4. At the begining, the batch algorithm with model selection (Algorithms 1 and 2) is called and the kernel width σ and initial estimate of parameters $\boldsymbol{\alpha}$ are calculated. Then, when a new sample $\boldsymbol{y}(t)$ (or, equivalently, a new sequence $\boldsymbol{Y}_{\mathrm{te}}(n_{\mathrm{te}}+1)$) is given, the reference and test intervals are shifted one step to the future (see Figure 2) and the estimate of parameters $\boldsymbol{\alpha}$ is updated using Algorithm 3. With the estimated parameters, the logarithm of the likelihood ratio (1) is evaluated as the change-detection score. If this score is beyond the given threshold μ (see Eq.(2)), the time point is reported as a change point and the current time t is updated as $t \leftarrow t + (n_{\mathrm{rf}} + n_{\mathrm{te}} + k)$. This procedure is repeated until the end of data.

4 Experimental Results

In this section, we experimentally investigate the performance of the proposed algorithm using artificial and real-world datasets. First, the proposed non-parametric direct density-ratio approach is compared with existing model-based likelihood-ratio approaches in Section 4.1. Then the proposed algorithm is compared with several popular change-point detection algorithms using real-world datasets in Section 4.2.

4.1 Artificial Dataset

Here we use the following three artificial datasets for illustration purposes.

```
   input: New sample $\boldsymbol{y}(t)$, the previous estimate of parameters $\boldsymbol{\alpha}$ and forgetting
          factors $\eta$ and $\lambda$.
1  Create the new sequence sample $\boldsymbol{Y}_{\mathrm{te}}(n_{\mathrm{te}}+1)$.
2  Update the parameters $\boldsymbol{\alpha}$:
       $\boldsymbol{\alpha} \leftarrow \begin{pmatrix} (1-\eta\lambda)\alpha_2 \\ (1-\eta\lambda)\alpha_3 \\ \vdots \\ (1-\eta\lambda)\alpha_{n_{\mathrm{te}}} \\ \eta/c \end{pmatrix},$
   where $c = \sum_{l=1}^{n_{\mathrm{te}}} \alpha_l K_\sigma(\boldsymbol{Y}_{\mathrm{te}}(n_{\mathrm{te}}+1), \boldsymbol{Y}_{\mathrm{te}}(l))$.
3  Perform feasibility satisfaction:
       $\boldsymbol{\alpha} \leftarrow \boldsymbol{\alpha} + (1 - \boldsymbol{b}^T\boldsymbol{\alpha})\boldsymbol{b}/(\boldsymbol{b}^T\boldsymbol{b}),$
       $\boldsymbol{\alpha} \leftarrow \max(\boldsymbol{0}, \boldsymbol{\alpha}),$
       $\boldsymbol{\alpha} \leftarrow \boldsymbol{\alpha}/(\boldsymbol{b}^T\boldsymbol{\alpha}),$
   where $b_l = \frac{1}{n_{\mathrm{rf}}} \sum_{i=1}^{n_{\mathrm{rf}}} K_\sigma(\boldsymbol{Y}_{\mathrm{rf}}(i), \boldsymbol{Y}_{\mathrm{te}}(l))$ for $l = 1, \ldots, n_{\mathrm{te}}$.
4  Update as $\boldsymbol{Y}_{\mathrm{rf}}(n_{\mathrm{rf}}+1) \leftarrow \boldsymbol{Y}_{\mathrm{te}}(1)$.
```

Algorithm 3: Online update of parameters $\boldsymbol{\alpha}$ in the Gaussian kernel model (3) in pseudo code.

```
    input: Window size $k$, interval lengths $n_{\mathrm{rf}}$ and $n_{\mathrm{te}}$, Gaussian width candidates
           $\{\sigma_i\}_{i=1}^{s}$, change-point detection threshold $\mu$, and, forgetting factors $\eta$ and $\lambda$.
1   Set $t_{\mathrm{rf}} = 1$ ($t = n_{\mathrm{rf}} + n_{\mathrm{te}} + k$), $\mathcal{Y}_{\mathrm{rf}}$ and $\mathcal{Y}_{\mathrm{te}}$.
2   Run the batch KLIEP (Algorithm 1) with $\mathcal{Y}_{\mathrm{rf}}$, $\mathcal{Y}_{\mathrm{te}}$ and $\sigma$ (in case of the first call, run
    the batch KLIEP with model selection (Algorithm 2)).
3   while not at the end of data do
4       $t \leftarrow t + 1$.
5       Update the estimate of parameters $\boldsymbol{\alpha}_t$ by Algorithm 3 with a new sample $\boldsymbol{y}(t)$ and
        the previous estimate $\boldsymbol{\alpha}_{t-1}$.
6       Compute the change-detection score $S$ (Eq.(1)).
7       if $S > \mu$ then
8           Report change at time $t$.
9           $t \leftarrow t + n_{\mathrm{rf}} + n_{\mathrm{te}}$ .
10          Go to step 2.
```

Algorithm 4: Pseudo code for the change-point detection algorithm based on the online density-ratio estimation.

Dataset 1 (see Figure 4 (a)):

The following autoregressive model (borrowed from the paper [37]) is used for generating 10000 time-series samples (i.e., $t = 1, \dots, 10000$):

$$y_t = 0.6y_{t-1} - 0.5y_{t-2} + \epsilon_t,$$

where ϵ_t is Gaussian noise with mean μ and standard deviation 1. The initial values are set as $y_1 = 0$ and $y_2 = 0$. We artificially create change points at every 1000 points: starting from $\xi = 0$, the noise mean is increased as $\xi \leftarrow \xi + i$ at time $t = 1,000 \times i$ for $i = 1, \dots, 9$.

Dataset 2 (see Figure 4 (b)):

The following autoregressive model (borrowed from the papers [28, 21]) is used for generating 1500 time-series samples:

$$y_t = e_t + \begin{cases} 0.97y_{t-1} + y_{t-2} - 0.5y_{t-3} + 0.97y_{t-4} & \text{if } t < 1000, \\ 0.97y_{t-1} + y_{t-2} - 0.7y_{t-3} + 0.97y_{t-4} & \text{if } t \geq 1000, \end{cases}$$

where e_t is Gaussian noise with mean 0 and standard deviation 0.2. The initial values are set as $y_1 = 0$, $y_2 = 8$, $y_3 = 6$ and $y_4 = 4$. The change point exists at $t = 1000$.

Dataset 3 (see Figure 4 (c)):

The following state-space model (borrowed from the paper [21]) is used for generating 1500 time-series samples:

$$\boldsymbol{x}_t = \boldsymbol{A}\boldsymbol{x}_{t-1} + \boldsymbol{u}_t,$$
$$y_t = C\boldsymbol{x}_{t-1} + v_t,$$

where $\boldsymbol{x}$ is a two-dimensional state vector, $\boldsymbol{u}$ is two-dimensional system noise following the uncorrelated Gaussian distribution with mean zero and variance $(0.5, 0.1)$, and v is scalar-valued observation noise following the Gaussian distribution with mean zero and variance 0.02. The parameters $\boldsymbol{A}$ and C are set as

$$\boldsymbol{A} = \begin{bmatrix} 0.95 & c \\ 0.1 & 1.0 \end{bmatrix} \text{ and } C = \begin{bmatrix} 0 & 1 \end{bmatrix},$$

where

$$c = \begin{cases} -0.3 & \text{if } t < 1000, \\ 0.05 & \text{if } t \geq 1000. \end{cases}$$

The change point exists at $t = 1000$.

For the proposed algorithm, the parameters are set as $k = 80$, $n_{\mathrm{rf}} = 100$, $n_{\mathrm{rf}} = 50$, $\eta = 1.0$ and $\lambda = 0.01$ for all three datasets. The threshold μ (see Eq.(2)) is set as

$$\mu = \begin{cases} 0.4 & \text{for the dataset 1,} \\ 0.2 & \text{for the dataset 2,} \\ 0.5 & \text{for the dataset 3.} \end{cases}$$

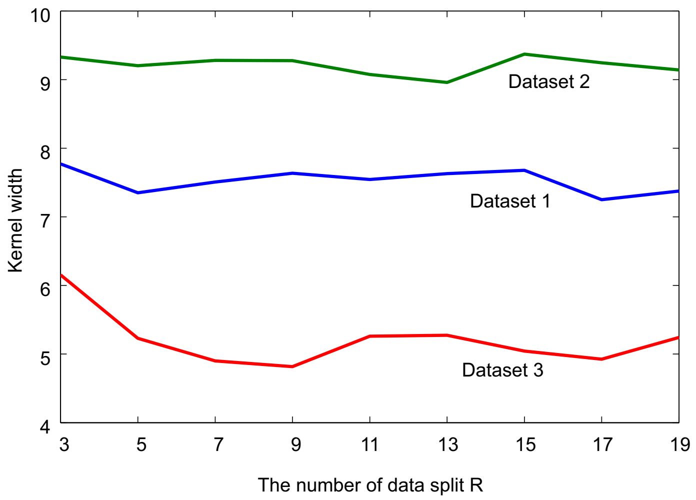


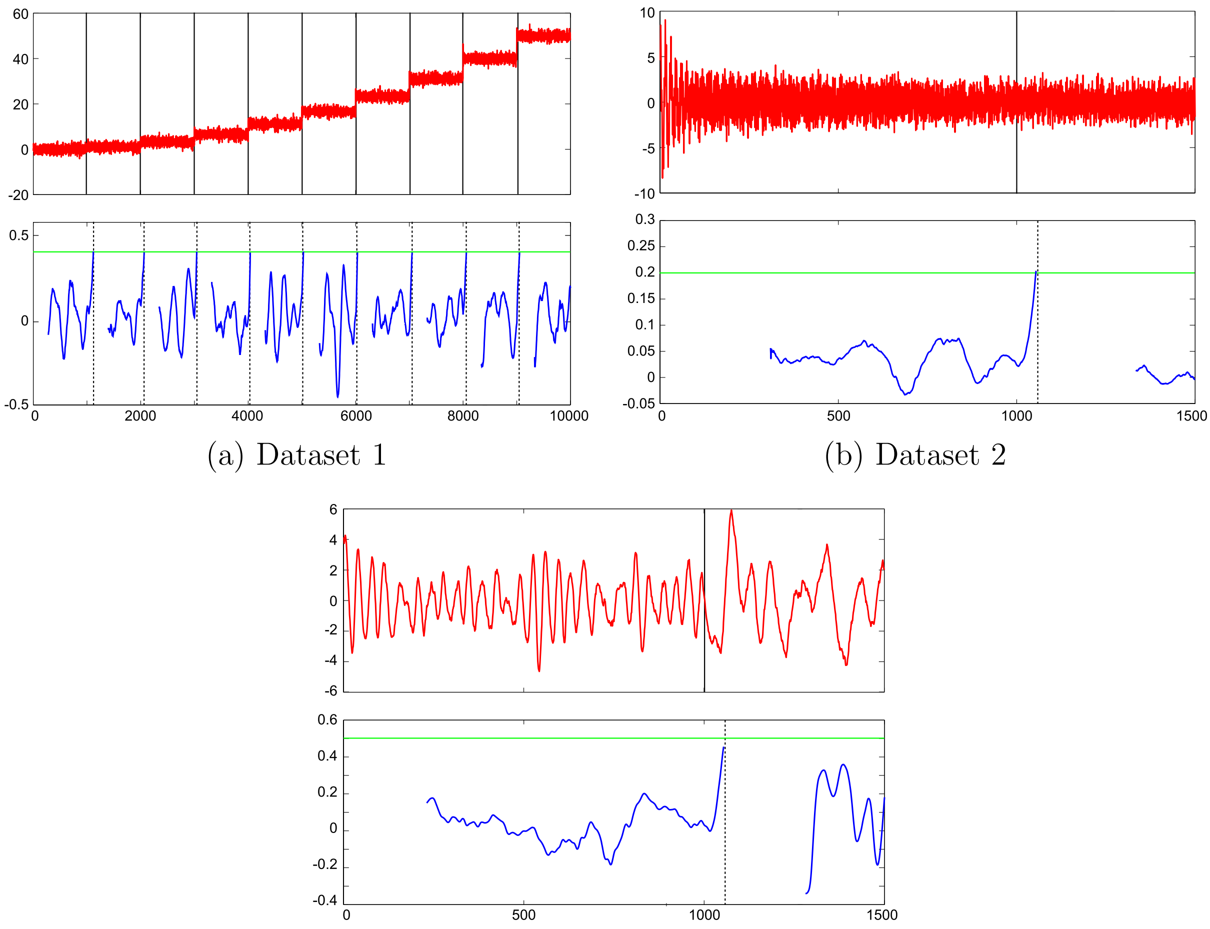
Figure 3: Averages of the kernel widths σ chosen by cross validation with several different number of data splits R for randomly generated 50 time-series with the models (Dataset 1, 2 and 3).

The kernel width σ in KLIEP is chosen by cross validation described in Algorithm 2. Firgure 3 shows the averages of the kernel widths σ chosen by cross validation with several different number of data splits R for randomly generated 50 time-series with the above models. The figure shows that the selected widths are not highly dependent on R. For this reason, we decided to use 5-fold cross validation in the rest of the experiments.

Figure 4 depicts time-series (upper graphs) samples as well as the change-detection score used in the proposed algorithm (i.e., the logarithm of the likelihood ratio S; see Eq.(1)) computed by the proposed method (lower graphs). The vertical black dotted lines in the graphs denote the detected change points (i.e., the change-detection score S goes beyond the threshold μ; which is denoted by the horizontal green solid lines in the graphs). The results show that after changes occur, the change-detection score of the proposed method increases rapidly and therefore the change points are detected accurately. Note that there is small delay in the detected changes—this is due to the data buffering process for constructing sequence samples. However, the amount of delay is systematic (in the current setup, $k + n_{\mathrm{te}} = 130$) and therefore the delay can be easily adjusted by substituting the length of the test interval from the time where the change is detected.

As shown above, the proposed method, in which the likelihood-ratio is estimated directly without going through density estimation in a non-parametric fashion, works reasonably well in the controlled setup. Next, we compare the performance of the proposed method with existing methods which are also based on the likelihood ratio but density estimation is explicitly involved in a parametric or non-parametric way.

Autoregressive (AR) model [37]: Time-series data is modeled by the following AR



(a) Dataset 1

(b) Dataset 2

(c) Dataset 3

Figure 4: Illustrative time-series samples (upper graphs) and the change-detection score obtained by the proposed method. The vertical black solid lines in the upper graphs denote the true change points. The vertical black dotted lines in the lower graphs denote the change points detected by the proposed method. The horizontal green solid lines denote the threshold for detecting change points.

model:

$$\boldsymbol{y}(t) = \sum_{i=1}^{b} A_i \boldsymbol{y}(t-i) + \boldsymbol{v}(t),$$

where $\{A_i\}_{i=1}^{b}$ are fixed parameters and $\boldsymbol{v}(t)$ is the Gaussian noise model with mean zero and variance parameter σ^2. The order b of the AR model is determined by cross validation. Here, two AR models are trained for modeling time-series in the reference and test intervals, and the logarithm of the ratio of the predictive densities of the last sample in the reference and test intervals is used as the change-detection score.

Kernel density estimation (KDE) [16]: KDE is a non-parametric technique to estimate a probability density $p(\boldsymbol{Y})$ from its i.i.d. samples $\{\boldsymbol{Y}(i)\}_{i=1}^{n}$. For the Gaussian kernel

(4), the KDE solution is given as

$$\widehat{p}(\boldsymbol{Y}) = \frac{1}{n(2\pi\sigma^2)^{dk/2}} \sum_{i=1}^{n} K_\sigma(\boldsymbol{Y}, \boldsymbol{Y}(i)),$$

where dk is the dimensionality of $\boldsymbol{Y}$ and $K_\sigma(\boldsymbol{Y}, \boldsymbol{Y}')$ is a kernel function. At each time step t, probability densities corresponding to reference and test intervals are estimated using KDE and then its logarithm of the likelihood ratio is used as the change-detection score.

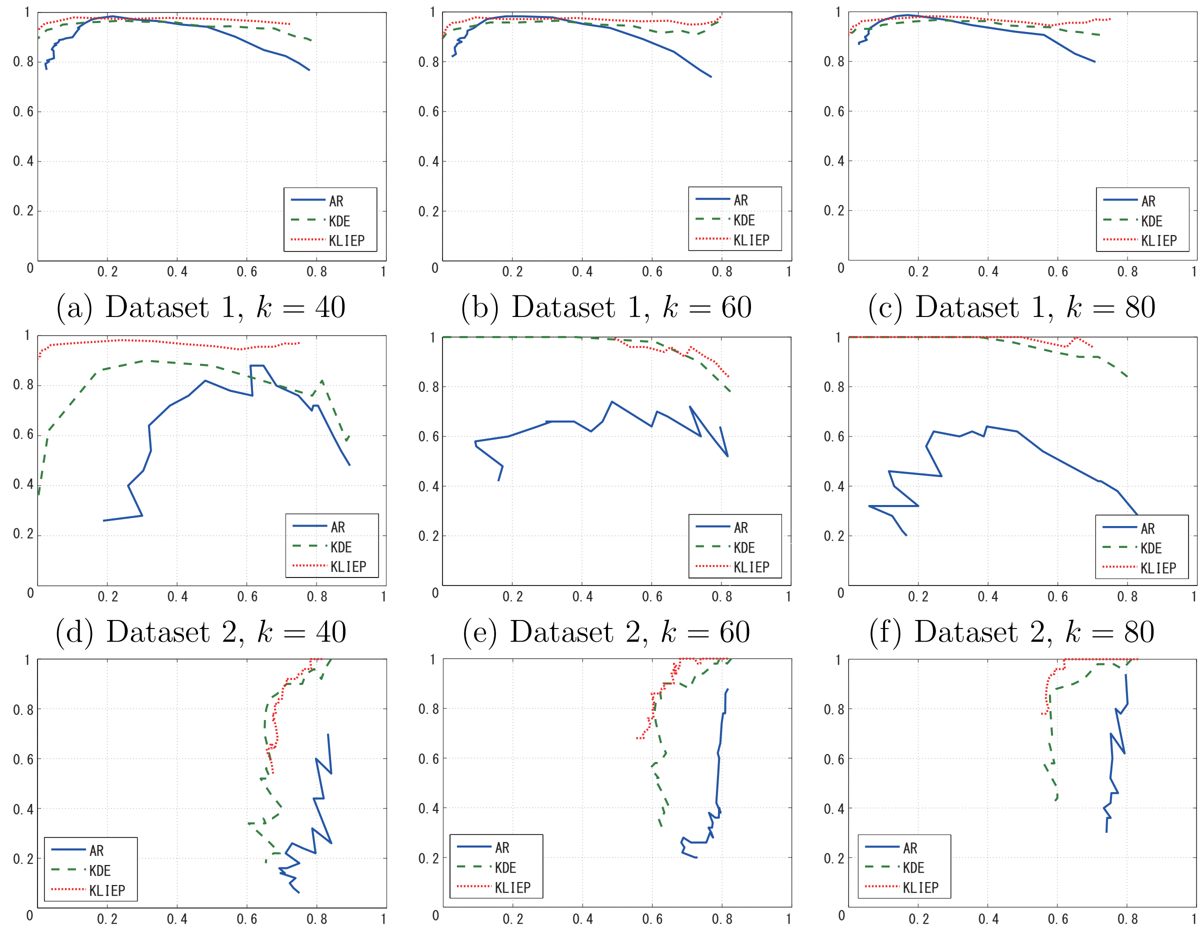
Figure 5 depicts the false-alarm rate versus accuracy (true positive) rate curves for several different values of the detection threshold μ for different window size k using randomly generated 50 time-series data with the above models.[2] More specifically, the horizontal axis of the graph corresponds to the false-alarm rate $n_{\mathrm{f}}/n_{\mathrm{al}}$ (n_{f} denotes the number of times non-change points are detected by mistake and n_{al} denotes the number of all detection alarms), and the vertical axis corresponds to the accuracy rate $n_{\mathrm{cr}}/n_{\mathrm{cp}}$ (n_{cr} denotes the number of times change points are correctly detected and n_{cp} denotes the number of all change points). Thus, a curve which is close to the left-upper corner means that the corresponding algorithm has good performance. Table 1 summarizes the best accuracy rate over the threshold μ for different window size k. More specifically, the table contains the maximum value of $n_{\mathrm{cr}}/n_{\mathrm{cp}}$ over μ and its corresponding accuracy degree $(n_{\mathrm{cr}} - n_{\mathrm{f}})/n_{\mathrm{cp}}$.

The results show that the AR model based method works well for the dataset 1 since the pre-specified AR model is correct. The AR model is still correct for the dataset 2, but the performance tends to be rather poor since the true data generation model is rather complex; thus it is difficult to choose the right order of the AR model using cross validation with a limited number of samples. The performance of the AR model based method also tends to be rather poor for the dataset 3 since the model is under-specified—the true data generation model is a state-space model which is substantially more complex than AR models.

The false-alarm rate versus accuracy rate curves of the KDE based and KLIEP based method are stably well for all the three datasets, thanks to the flexibility of non-parametric estimation. However, since non-parametric density estimation is a hard task and tends to be rather inaccurate when only a limited number of samples is available, the performance of the KDE based methods can be degraded. On the other hand, the KLIEP based method can avoid this difficulty by directly estimating the density ratio without going through density estimation. Thus it tends to perform better. Indeed, the curves of the KLIEP based method tends to be better than those of the KDE based method for the datasets 1 and 2; they are comparable for the dataset 3.

Another important finding from the experimental results is that the performance of all the methods tend to be degraded when the window size k is too small. This is because time-correlation of the series-samples is still strong for a small k and therefore it is not appropriate

[2] A false-alarm vs. accuracy rates curve is usually called an ROC (Receiver Operating Characteristic) curve. However, in our case, i.e., sequential change-point detection, the number of samples used in calculating the rates is not uniform at each point on the curve because it depends on the number of detected change points. Thus, the curves in Figure 5 may not have the properties that usual ROC curves have, e.g., monotone non-decreasing.



(a) Dataset 1, $k = 40$ (b) Dataset 1, $k = 60$ (c) Dataset 1, $k = 80$

(d) Dataset 2, $k = 40$ (e) Dataset 2, $k = 60$ (f) Dataset 2, $k = 80$

(g) Dataset 3, $k = 40$ (h) Dataset 3, $k = 60$ (i) Dataset 3, $k = 80$

Figure 5: False-alarm rate versus accuracy rate curves for the artificial datasets for different window size k.

to regard them as independent. Thus our strategy of using subsequence samples for change detection would be a reasonable choice.

4.2 Real-World Dataset

As shown above, the proposed method works reasonably well for the illustrative change-detection problems. Next, we apply the proposed method to real-world datasets and investigate whether the good performance is still obtained in realistic circumstances.

In this experiment, we compare the performance of the proposed method with the following four algorithms:

Singular-spectrum analysis (SSA) [28]: SSA evaluates the degree of changes between two consecutive sequences using the distance defined based on singular spectrums.

Change finder (CF) [36, 37]: CF first sequentially fits an AR model to time-series data

Table 1: The best accuracy rate and its accuracy degree for the artificial datasets. The results of the best methods are described in bold face.

(a) Accuracy rate

	k	AR	KDE	KLIEP
1	40	**98.4**	96.7	98.0
	60	**98.2**	96.4	97.8
	80	**98.7**	96.7	98.2
	Avg.	**98.4**	96.6	98.3
2	40	88.0	90.0	**98.2**
	60	74.0	**100.0**	**100.0**
	80	62.0	**100.0**	**100.0**
	Avg.	74.7	96.7	**99.4**
3	40	70.0	**100.0**	**100.0**
	60	88.0	**100.0**	**100.0**
	80	94.0	**100.0**	**100.0**
	Avg.	84.0	**100.0**	**100.0**

(b) Accuracy degree

	k	AR	KDE	KLIEP
1	40	77.0	72.7	**87.6**
	60	80.1	63.5	**88.3**
	80	**81.8**	71.7	74.4
	Avg.	79.6	69.3	**83.4**
2	40	26.9	59.2	**74.4**
	60	25.4	**100.0**	**100.0**
	80	37.6	**100.0**	**100.0**
	Avg.	30.0	86.4	**91.5**
3	40	-13.2	15.9	**21.3**
	60	6.3	20.1	**32.1**
	80	14.4	18.7	**38.2**
	Avg.	7.5	18.2	**30.5**

and auxiliary time-series data is generated from the AR model. Then another AR model is fitted to the auxiliary time-series and its log-likelihood is used as the change-detection score.

Subspace identification (SI) [21]: SI identifies a subspace in which time-series sequences are constrained and evaluates the distance of target sequences form the subspace. A recursive subspace identification algorithm based on an instrumental variable method is used for estimating the subspace.

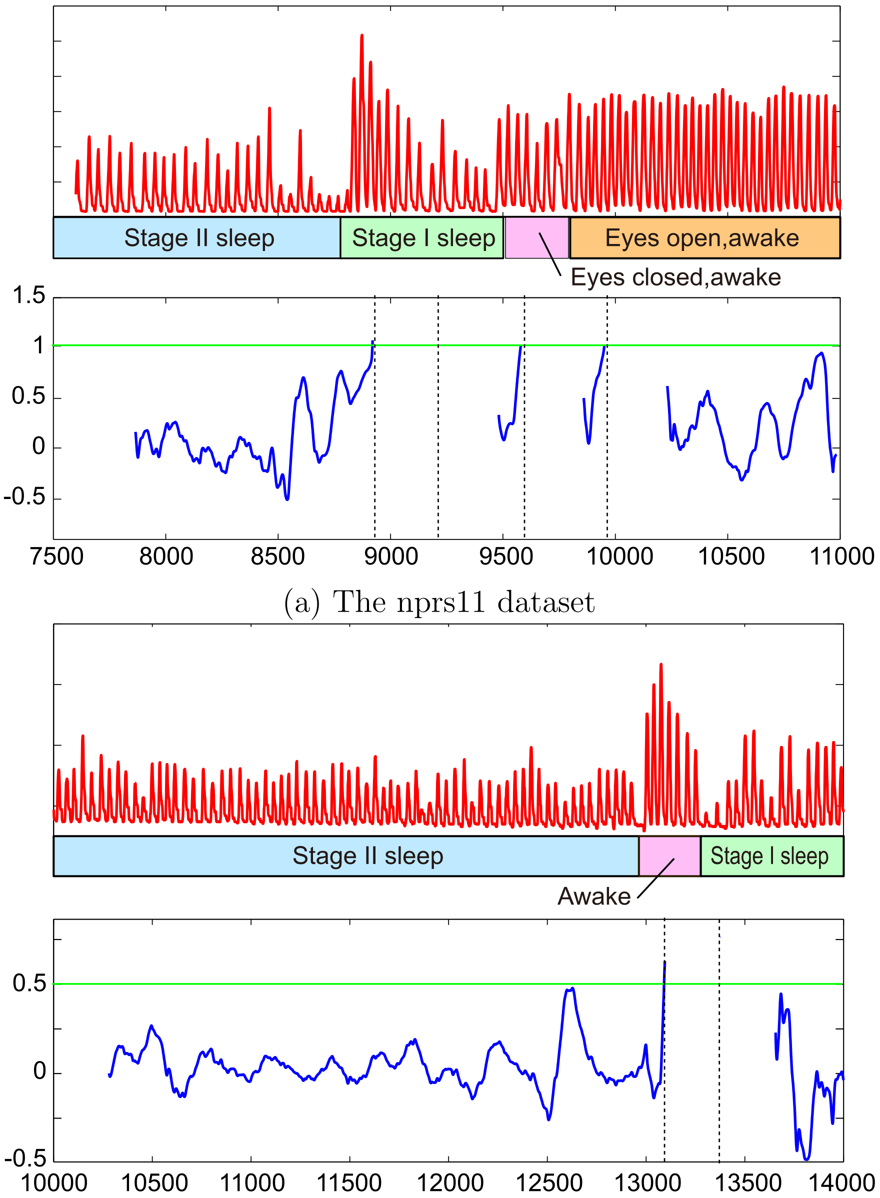
One-class support vector machine (OSVM) [11]: This algorithm iteratively compares the supports of probability densities estimated by one-class support vector machine from samples in the reference and test intervals.

Here, two different kinds of datasets, namely, the respiration and speech datasets are used in Section 4.2.1 and in Section 4.2.2, respectively.

4.2.1 Respiration Dataset

The first real-world dataset we use here is the *respiration dataset* in the UCR Time Series Data Mining Archive[3]. This dataset contains 15 time-series data—each of which records patients' respiration measured by thorax extension and every time period is manually annotated by a medical expert as '*awake*', '*sleep*' etc. [23]. Two examples of the original time-series as well as the annotation results are depicted in the top graphs of Figure 6. The task is to detect the time points at which the state of patients changes.

[3] Available from '`http://www.cs.ucr.edu/∼eamonn/discords/`'.



(a) The nprs11 dataset

(b) The nprs43 dataset

Figure 6: The raw time-series data of the respiration datasets (top) and the change-detection score (1) of the proposed method (bottom).

Figure 6 illustrates the original time-series (upper) as well as the change-detection score (see Eq.(1)) of the proposed method (bottom). The change points detected for

$$\mu = \begin{cases} 1.0 & \text{for the nprs11 dataset,} \\ 0.5 & \text{for the nprm43 dataset,} \end{cases}$$

are indicated by the vertical dotted lines. We set $k = 80$, $n_{\mathrm{rf}} = 100$, and $n_{\mathrm{te}} = 100$, and the kernel width σ is chosen by likelihood cross validation using Algorithm 2. The graphs

Table 2: Accuracy rate (top) and degree (bottom) of change detection for the respiration dataset. The results of the best methods are described in bold face.

(a) Accuracy rate

	CF	SSA	SI	OSVM	KLIEP
$k = 30$	22.1	46.5	47.5	49.7	**49.8**
$k = 50$	25.2	46.5	47.9	49.6	**50.0**
Average	23.7	46.5	47.7	49.7	**49.9**

(b) Accuracy degree

	CF	SSA	SI	OSVM	KLIEP
$k = 30$	-59.0	-40.0	-42.3	-44.3	**-38.4**
$k = 50$	-54.5	-38.2	-39.4	-44.2	**-35.3**
Average	-56.8	-39.1	-40.9	-44.2	**-36.9**

show that, although the increase of the change-detection score is not so clear as that in the illustrative examples in Figure 4, the proposed algorithm still works reasonably well for the real-world respiration datasets. Note that in the nprs11 dataset, the time point around 9250 is detected incorrectly.

The comparison results are summarized in Table 2 (the best accuracy rate and its corresponding accuracy degree). All the results are averaged over the 15 datasets. The table shows that the best performance of the proposed method is better than the other methods.

4.2.2 Speech Dataset

The second real-world dataset is the *speech dataset* CENSREC-1-C in the Speech Resource Consortium (SRC) corpora provided by National Institute of Informatics (NII) [4]. This dataset contains dozens of time-series data – each of which records speech signals measured with microphones in several noise environments and the begining and ending of every speech are manually annotated. One example of the original time-series data as well as the annotation results are depicted in the top graph of Figure 7. The task here is to detect the time points at which a speech begins or ends. Before feeding the data to the algorithms, we took the moving average over 10 samples for smoothing purposes.

Figure 7 illustrates a part of the original time-series data (upper) as well as change-detection score (see Eq.(1)) of the proposed method (bottom). The change points detected for $\mu = 1.3$ are indicated by the vertical dotted lines. We set $k = 100$, $n_{\mathrm{rf}} = 200$ and $n_{\mathrm{te}} = 100$, and the kernel width σ is chosen by likelihood cross validation using Algorithm 2. The graphs show that the proposed algorithm still works reasonably well also for the real-world speech dataset. Note that the time points around 22200 and 28400 are detected incorrectly, and the change point around 28400 is not detected by the proposed method in

[4] Available from '`http://research.nii.ac.jp/src/eng/index.html`'.

this case.

The comparison results are summarized in Table 3 (the best accuracy rate and its corresponding accuracy degree). All results are averaged over the first 10 datasets (in the STREET_SNR_HIGH dataset). The table shows that the best performance of the proposed method is also competitive with the best existing algorithms.

Overall, we experimentally confirmed that the proposed method could be a useful alternative to the existing methods in practical change detection applications.

5 Conclusions and Future Prospects

We formulated the problem of change-point detection in time-series data as the problem of comparing probability distributions over two consecutive time intervals that generate the time-series data. Within this framework, we proposed a novel non-parametric algorithm for change-point detection. The key idea of the proposed method is that the *ratio* of two probability densities is directly estimated without going through density estimation. Thus, the proposed method can avoid non-parametric density estimation, which is known to be a hard problem in practice. We also derived an online algorithm that can update the density ratio estimator efficiently based on the previous estimation result. We experimentally showed the usefulness of the proposed algorithm using artificial and real-world datasets.

In our framework, the dimensionality of data samples tends to be high because not observations at each time point but *sequences* of observations are treated as samples for estimation. This can potentially cause performance degradation in density-ratio estimation. A possible measure for this is to incorporate some dimensionality reduction scheme into the density-ratio estimation framework [33, 35]. We will apply such dimensionality reduction methods to change-detection problems in the future work.

Another important issue to be further investigated is model selection—the performance of change-point detection depends on the selection of the kernel width σ. In the current implementation of the proposed method, the width is chosen by cross validation in the beginning and the same fixed width is used over the entire online learning process. Although the current implementation showed a superior performance in experiments, its performance could be further improved if the kernel width is optimized through the online learning process. Another challenging future work along this line would be to develop a computationally efficient online density-ratio estimation method that can perform model selection also in an online manner.

In our method, there are several tuning parameters other than σ, such as, window size k, interval lengths n_{rf} and n_{te}, and forgeting factors η and λ. Basically, the paremeter k does not affect the performance so much if it is sufficiently large (≥ 40), as can be observed in the experiments. As for n_{rf} and n_{te}, the larger these are, the more accurate the estimation of density-ratio would be, as common statistical algorithms. However, since the interval requred for initialization becomes longer with increasing of n_{rf} or n_{te}, too large n_{rf} or n_{te} may make the method less practical for time-series data where changes occur frequently. η and λ control effects from past data by gradually decreasing the weights on past time points,

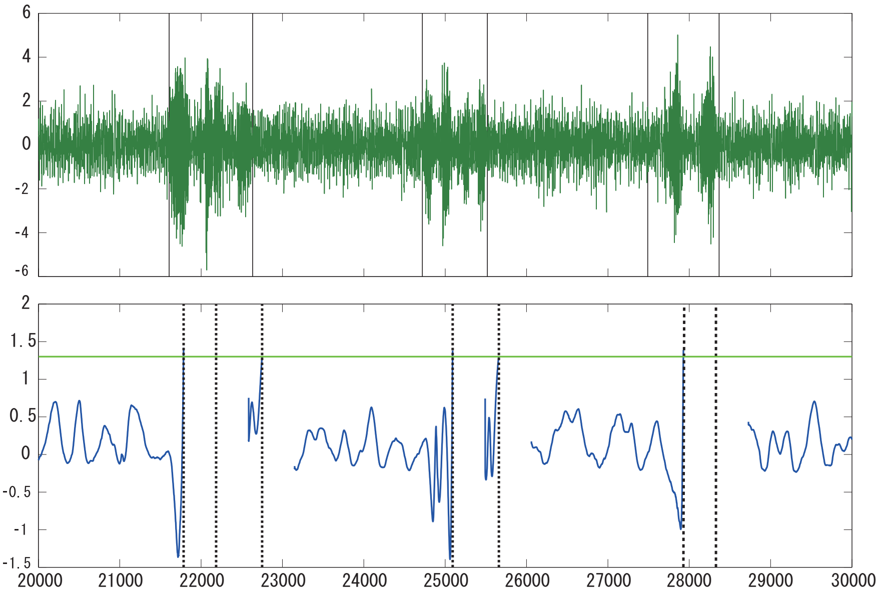


Figure 7: The raw time-series data of the speech dataset (top) and the change-detection score (1) of the proposed method (bottom). The vertical black solid lines in the top graph denote the true change points. The vertical black dotted lines in the bottom graph denote the change points detected by the proposed method. The horizontal green solid lines denote the threshold for detecting change points

Table 3: Accuracy rate (top) and degree (bottom) of change detection for the speech dataset. The results of the best methods are described in bold face.

(a) Accuracy rate

	CF	SSA	SI	OSVM	KLIEP
$k = 50$	34.0	**43.5**	39.0	**43.5**	42.5
$k = 100$	39.0	45.0	40.5	**53.0**	45.0
Average	36.5	44.3	39.8	**47.8**	43.8

(b) Accuracy degree

	CF	SSA	SI	OSVM	KLIEP
$k = 50$	-54.3	**-34.6**	-35.0	-48.4	-34.7
$k = 100$	-47.6	**-24.4**	-25.8	-35.8	-28.1
Average	-51.0	**-29.5**	-30.4	-37.1	-31.4

which is commonly used heuristics when dealing with non-stationary time-series data. When stationarity can be assumed, we can just set these parameters to zeros. Currently, these parameters need to be chosen beforehand. Our method would be more useful if these tuning parameters are selected automatically only from data, which is another future work.

Acknowledgements

This research was supported in part by JSPS Global COE program "Computationism as a Foundation for the Sciences", Grant-in-Aid (22700147) from the Ministry of Education, Culture, Sports, Science and Technology, the JST PRESTO program, SCAT, AOARD and JFE 21st Century Foundation.

References

[1] S. Adak. Time-dependent spectral analysis of nonstationary time series. *Journal of the American Statistical Association*, 93(444):1488–1501, 1998.

[2] S. Amari. Theory of adaptive pattern classifiers. *IEEE Transactions on Electronic Computers*, EC-16(3):299–307, 1967.

[3] V. Guralnik J. and Srivastava. Event detection from time series data. In *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'99)*, pages 33–42, 1999.

[4] J. Bai and P. Perron. Computation and analysis of multiple structural change models. *Journal of Applied Econometrics*, 18(1):1–22, 2003.

[5] M. Basseville and V. Nikiforov. *Detection of Abrupt Changes: Theory and Application.* Prentice-Hall, Inc., Englewood Cliffs, N. J., 1993.

[6] R. J. Bolton and D. J. Hand. Statistical fraud detection: A review. *Statistical Science*, 17(3):235–255, 2002.

[7] B. Brodsky and B. Darkhovsky. *Nonparametric Methods in Change-Point Problems.* Kluwer Academic Publishers, 1993.

[8] H. Choi, H. Ombao, and B. Ray. Sequential change-point detection methods for nonstationary time series. *Technometrics*, 50(1):40–52, 2008.

[9] M. Csörgö and L. Horváth. Nonparametric methods for changepoint problems. In P. R. Krishnaiah and C. R. Rao, editors, *Quality and Control and Reliability, Handbook of Statistics*, volume 7, pages 403–425. 1988.

[10] R. A. Davis, T. C. M. Lee, and G. A. Rodriguez-Yam. Structural break estimation for nonstationary time series models. *Journal of the American Statistical Association*, 101(473):223–239, 2006.

[11] F. Desobry, M. Davy, and C. Doncarli. An online kernel change detection algorithm. *IEEE Transactions on Signal Processing*, 53(8):2961–2974, 2005.

[12] G. S. Fishman. *Monte Carlo: Concepts, Algorithms, and Applications*. Springer-Verlag, Berline, 1996.

[13] R. Fujimaki, T. Yairi, and K. Machida. An approachh to spacecraft anomaly detection problem using kernel feature space. In *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 401–410, 2005.

[14] F. Gustafsson. The marginalized likelihood ratio test for detecting abrupt changes. *IEEE Transactions on Automatic Control*, 41(1):66–78, 1996.

[15] F. Gustafsson. *Adaptive Filtering and Change Detection*. John Wiley & Sons Inc., 2000.

[16] W. Härdle, M. Müller, S. Sperlich, and A. Werwatz. *Nonparametric and Semiparametric Models*. Springer Series in Statistics. Springer, Berlin, 2004.

[17] J. Huang, A. Smola, A. Gretton, K. M. Borgwardt, and B. Schölkopf. Correcting sample selection bias by unlabeled data. In *Advances in Neural Information Processing Systems*, volume 19. MIT Press, Cambridge, MA, 2007.

[18] T. Ide and H. Kashima. Eigenspace-based anomaly detection in computer systems. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'04)*, pages 440–449, 2004.

[19] T. Kanamori, T. Suzuki, and M. Sugiyama. Theoretical analysis of density ratio estimation. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E93-A(4):787–798, 2010.

[20] T. Katayama. *Subspace Methods for System Identification: A Realization Approach*. Communications and Control Engineering. Springer Verlag, 2005.

[21] 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 (ICDM'07)*, pages 559–564, Omaha, NE, 2007.

[22] Y. Ke, R. Sukthankar, and M. Hebert. Event detection in crowded videos. In *Proceedings of the 11th IEEE International Conference on Computer Vision (ICCV'07)*, pages 1–8, 2007.

[23] E. Keogh, J. Lin, and A. Fu. HOT SAX: Efficiently finding the most unusual time series subsequences. In *Proceedings of the 5th IEEE International Conference on Data Mining (ICDM'05)*, pages 226–233, 2005.

[24] D. Kifer, S. Ben-David, and J. Gehrke. Detecting change in data streams. In *Proceedings of the 30th International Conference on Very Large Data Bases (VLDB'04)*, pages 180–191, 2004.

[25] J. Kivinen, A. J. Smola, and R. C. Williamson. Online learning with kernels. *IEEE Transactions on Signal Processing*, 52(8):2165–2176, 2004.

[26] M. Markou and S. Singh. Novelty detection: A review - part 1: Statistical approaches. *Signal Processing*, 83(12):2481–2497, 2003.

[27] M. Markou and S. Singh. Novelty detection: A review - part 2: Neural network based approaches. *Signal Processing*, 83(12):2499–2521, 2003.

[28] V. Moskvina and A. A. Zhigljavsky. An algorithm based on singular spectrum analysis for change-point detection. *Communication in Statistics: Simulation & Computation*, 32(2):319–352, 2003.

[29] U. Murad and G. Pinkas. Unsupervised profiling for identifying superimposed fraud. In *Proceedings of the 3rd European Conference Principles and Practice of Knowledge Discovery in Databases (PKDD'99)*, pages 251–261, 1999.

[30] H. C. Ombao, J. A. Raz, R. von Sachs, and B. A. Malow. Automatic statistical analysis of bivariate non-stationary time series. *Journal of the American Statistical Association*, 96(454):543–560, 2001.

[31] P. V. Overschee and B. D. Moor. *Subspace Identification for Linear Systems: Theory, Implementation and Applications.* Kluwer Academic Publishers, Dordrecht, 1996.

[32] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1471, 2001.

[33] M. Sugiyama, M. Kawanabe, and P. L. Chui. Dimensionality reduction for density ratio estimation in high-dimensional spaces. *Neural Networks*, 23(1):44–59, 2010.

[34] M. Sugiyama, T. Suzuki, S. Nakajima, H. Kashima, P. von Bünau, and M. Kawanabe. Direct importance estimation for covariance shift adaptation. *Annals of the Institute of Statistical Mathematics*, 60(4), 2008.

[35] M. Sugiyama, M. Yamada, P. von Bünau, T. Suzuki, T. Kanamori, and M. Kawanabe. Direct density-ratio estimation with dimensionality reduction via least-squares hetero-distributional subspace search. *Neural Networks*, 2011.

[36] Y. Takeuchi and K. Yamanishi. A unifying framework for detecting outliers and change points from non-stationary time series data. *IEEE Transactions on Knowledge and Data Engineering*, 18(4):482–489, 2006.

[37] K. Yamanishi and J. Takeuchi. A unifying framework for detecting outliers and change points from non-stationary time series data. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'02)*, pages 676–681, 2002.

[38] K. Yamanishi, J. Takeuchi, G. 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.