Article

Kernel Complexity for Nonparametric Distributions and Detection of Its Changes

So Hirai 1* and Kenji Yamanishi 2

1 Graduate School of Information Science and Technology, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo, Japan; so_hirai@mist.i.u-tokyo.ac.jp
2 Graduate School of Information Science and Technology, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo, Japan; yamanishi@mist.i.u-tokyo.ac.jp
* Correspondence: so_hirai@mist.i.u-tokyo.ac.jp

Abstract: This study is concerned with the quantification of structural information in the form of an index for nonparametric distributions with the aim of detecting their changes in time series data. In a parametric model such as a Gaussian mixture model, the number of clusters can represent the structural information. However, the notion of structural information for modeling data nonparametrically does not exist. Herein we introduce the novel notion of kernel complexity (KC) as structural information in a nonparametric setting. The key idea of KC is to combine the information bias by the Gini index with the quantity of information measured by the normalized maximum likelihood (NML) code length. We empirically show the similarities between KC and the number of clusters in a parametric model under certain conditions. We propose an algorithm to detect structural change with KC in nonparametric distributions. Using synthetic and real datasets, we empirically demonstrate that our framework enables us to detect the structural changes underlying the data. We use the synthetic datasets to demonstrate the usefulness of our method with some characteristic data distribution. We then use real datasets to evaluate the validity of the detected results both quantitatively and qualitatively.

Keywords: change-point detection; kernel density estimation; normalized maximum likelihood (NML); minimum description length (MDL) principle; kernel complexity (KC).

1. Introduction

1.1. Motivation

In this study, we define new structural information related to a nonparametric distribution and propose a method to detect the changes in this distribution over time. The structure referred to here is, for example, a cluster when a parametric model is employed for clustering purposes. The number of clusters in a clustering model expresses structural information as a group of aggregated data points. In contrast, as it is not possible to define a cluster in a nonparametric model, statistics have previously been used to capture structures using the method of moments. However, even in nonparametric distributions, it is important to aggregate structural information in the form of data-like clusters to allow a global understanding of the data. In this study, we propose a new index for structural information. This index, which we termed the kernel complexity (KC), defines the structural information for a nonparametric model. The index is defined by using the Gini index to measure the density of data in terms of information bias [1]. We measure the amount of information provided by the data by determining the minimum description length (MDL), i.e., the shortest code length required to encode the data into a binary sequence in a prefix coding method. This is because
of the MDL principle [2], which asserts that the best way to model data is to select a model that compresses the data and the model itself. Furthermore, we propose an algorithm to detect changes in the KC when the data are in the form of a time series. This algorithm provides a framework for the detection of changes based on the KC.

The data targeted in this research are provided as a time series, and we consider data that consist of many multidimensional data points for each point in time. Situations that give rise to temporal data such as these are e.g., customer purchases and sensor measurements. The ability to meaningfully analyze these data would be useful for marketing purposes, for example. In the domain of marketing, suppose we are given a sequence of datasets that contain a number of patterns of purchase behavior that were observed over a period of time. Because user behavior changes as time passes, different data distributions would be expected to appear as a result of the users’ actions. A dataset of which the structure changes as a function of time is illustrated in Figure 1.

This study is concerned with detecting the changes the aforementioned data undergo without assuming a specific distribution. Because we do not assume a specific parametric model, we consider an approach that captures the changes in the distribution rather than observing the changes with respect to the number of clusters. For this purpose, we use kernel density estimation as a way to represent the data distribution. This approach enables us to even handle complex data chunks that cannot be represented by parametric models. We calculate the index, KC, by making some assumptions (e.g., by restricting the parameter on the normalized maximum likelihood (NML) calculation, by restricting data structure, etc.). When capturing the changes in the distribution itself, we apply the concept of clustering in a parametric model and propose a method to express the complexity of the structure of the dataset in terms of the density of the dataset. In this case, the amount of information used as a reference when capturing the overall image of the distribution would be expected to differ between the high- and low-density sections of the dataset. Indexing the degree of deviation of this information would then enable us to determine whether it is a complex distribution (a state in which the dataset is sparsely distributed) or a plain distribution (a state in which the dataset is densely distributed). It is possible to detect changes in the density distribution of data by observing an index in a time series. For the purposes of this work, we formulate the following hypotheses for considering changes in the data distributions.

- At a time when there is no change, the distribution maintains a certain form, and the index is not observed to change.
- When the distribution changes, this change is considered to occur as a gradual transformation from the original shape until it finally stabilizes as the changed shape. In the period during which the change is undergoing transition, the index is also considered to gradually change owing to the change in the distribution.

Under these hypotheses, we propose an index that quantifies the degree of information bias. In addition, we propose a method that uses this index to detect the change from the perspective of the distribution density.

This study has three aims. The first aim is to propose an index, named the KC, which defines the structural information of a dataset. This index is defined as an indicator of the distribution of peaks in the dataset similar to a cluster in clustering. The second aim is to develop an algorithm for detecting changes in structure in a sequential setting. The third is to demonstrate the usefulness of the proposed method by using two types of datasets, i.e., artificial and real practical datasets.

1.2. Related Work

We consider the related work from the viewpoints of both structural information and change detection. To define structural information in terms of the density, the use of clustering methods (i.e., aggregating data points into subgroups) is a well-known approach in parametric models (e.g., [3], [4]). These clustering methods are able to determine the optimal number of clusters [5–7]. The
A chunk of a dataset changes over time.

Figure 1. Change in the structure of a dataset over time. The data points gradually move, causing the distribution of the dataset to gradually change.

The moment method is a well-known method for defining the stochastic features to ultimately define the structure of a nonparametric distribution [8]. Furthermore, the Kolmogorov complexity\(^1\) [9] has long been used as an index to indicate the complexity of data sequences. With reference to nonparametric approaches for model selection, Zhang and Kosecká proposed a method to find the number of regression lines [10]. A method that determines the number of clusters with kernel k-means was proposed by Kyrgyzov et al. [11]. This method uses the information related to a latent variable known as the cluster index. Our approach is to aim to quantify the global features of a distribution without explicitly using the concept of clustering.

From the viewpoint of change detection, the changes undergone by models or parameters have been detected by using various models that have been developed for this purpose. In the context of parametric models, Herbster et al. proposed a method for tracking the best expert. This method sequentially updates the weights of the models [12]. Erven et al. proposed the concept of a switching distribution and a method for selecting a model in a situation in which the model changes over time [13]. Yamanishi and Maruyama proposed a dynamic model selection (DMS) algorithm based on the MDL principle to detect the changes in statistical models [14,15]. As an extension of this algorithm, Hirai and Yamanishi proposed the sequential DMS (SDMS) algorithm, which is used to apply DMS to a sequential setting [16]. Spiliopoulou et al. conducted research that adds a qualitative consideration to the change in clustering by proposing a method to identify the nature of the transition of a cluster on the basis of data movement between clusters [17].

Methods that use nonparametric models have also been proposed. In the supervised scenario, various methods based on the idea of concept drift have been developed [18]. Liu et al. proposed a method based on estimation of the direct density ratio [19], and Jeske proposed a method referred to as the CUSUM algorithm, which detects changes using the cumulative sum [20]. Tan et al. proposed Bayesian change-point detection [21], which is a method based on the probability density. As a nonparametric approach targeted by this research, Harchaoui et al. proposed an algorithm that employs the kernel Fisher discriminant ratio [22], and Saatçi et al. proposed a method based on a Gaussian process [23]. All of these methods detect changes based on the extent to which the data distribution differs from that at a previous point in time. Our aim is to detect changes with respect to the data aggregation by proposing changes in the values that quantify the structural information in terms of the density.

\(^1\) The Kolmogorov complexity is formally incomputable in general.
1.3. Significance of This Study

The significance of this study is summarized as follows:

1. **The proposal of an index to ascertain structural information without a specific probability distribution**

   We propose an index named KC to ascertain the structural information of aggregated data. The KC is defined by measuring the density of a dataset by determining the information bias with a Gini index; a larger KC indicates that the distribution of the dataset is wider and the structure is more complex. Rather than being a discrete value that indicates the number of clusters in a parametric model, the KC is an index that takes continuous values. We can use KC as a new quantitative indicator to ascertain nonparametric global information.

   When using the KC, we especially use the NML code length based on the MDL principle as a criterion to express information. The reason why we employ the NML code length is that it has attractive properties:

   - (a) it minimizes Shtarkov’s minimax risk [24].
   - (b) it optimizes the minimax estimation [25].

   Even though we adopt the kernel density estimation as a nonparametric density estimation method, its NML code length cannot be directly calculated. This means we need to consider two main points when calculating the NML code length. The first is to introduce a subprobability distribution associated with the kernel density estimation. The second is to propose a method for calculating the NML for this subprobability distribution by introducing a previously proposed concept [26].

2. **The proposal of a method that uses KC to detect the points of structural change**

   In this paper, we propose an algorithm to detect the changes in KC when the data are given as a time series. This makes it possible to detect changes in the global structure of nonparametric data.

   Many change-point detection algorithms for nonparametric distributions have been proposed (e.g., [22,23]). In comparison, this algorithm only detects changes in the global information measured by KC and provides a new view of nonparametric change detection. In addition, KC does not always change suddenly; instead, it may experience gradual change. Because KC is a continuous value, it is also able to detect such gradual change effectively. The use of KC enables us to rank the time of interest by adding the degree of change to non-stationary time series data.

3. **Empirical validation of the usefulness of the proposed algorithm**

   In this study, we use simulation to empirically demonstrate the usefulness of the proposed algorithm. In a simulation using artificial datasets, we find that the proposed algorithm can calculate the degree of change and detect changes. In relation to the benefit, delay, false alarm rate (FAR), and area under the curve (AUC) of the evaluation, the proposed algorithm was found capable of detecting points of change earlier compared to other methods.

   We empirically demonstrate that our algorithm can detect the changes in an actual dataset. That is, our algorithm successfully processed: a dataset containing electricity consumption values to detect high complexity at the time at which a change in lifestyle occurred; a dataset consisting of sensor measurements to detect meaningful changes in the electric power consumption; a dataset containing data pertaining to beer purchases to detect changes in customers’ purchasing behavior.
2. Criterion for Complexity of Dataset

2.1. Problem Setting

We consider a situation in which the distribution of the dataset $X_t$, which is observed at each time $t$, gradually changes over time. Each dataset $X_t$ can be expressed as $X_t = x^n = (x_1, \ldots, x_n)^\top \in \mathbb{R}^{n \times m}$, which consists of $n$ data points of dimension $m$. For this purpose, we measure the density variation as a complexity and detect change by detecting the change in density.

In this study, $X_t$ is considered to have a complex distribution. Thus, we aim to define the structure of the dataset without using a specific parametric distribution. In this section, for brevity, we express $X_t$ as $X = x^n = (x_1, \ldots, x_n)^\top \in \mathbb{R}^{n \times m}$. As we do not assume a specific parametric distribution, we use the following kernel density estimation:

$$ f(x; h) = \frac{1}{n} \sum_{j=1}^{n} K(x - x_j; h), $$

where $K(\cdot)$ is the kernel function, and we use the following Gaussian kernel:

$$ K(x; h) = \frac{1}{(2\pi h^2)^{m/2}} \exp \left( -\frac{|x|^2}{2h^2} \right). $$

2.2. Concept of the Proposed Method

Here we describe the concept of the proposed method. The degree of concentration of a dataset at each time is expressed by focusing on the amount of information contained in each data point included in the dataset. The amount of information represents the information associated with each data point in defining a distribution. For example, we consider the density distribution in Figure 2a. The density distribution indicated by the solid blue line is expected to be defined by two high-density peaks. Conversely, the structure of the density distribution indicated by the dotted orange lines can be considered to be necessary to observe the structure of the density distribution by taking a wide view of the data space. To understand the structure of the density distribution with the help of the diagrams in the figure, an indicator would be useful to define a distribution as a complex distribution when the information is widely distributed by measuring the amount of information contained in the dense and sparse parts. For a sufficiently high density, we can think of an indicator that defines a distribution as a dense (relatively easy) distribution. Then, we derive an index of the structural information termed KC.

We measure the amount of information of the data in terms of MDL, i.e., the shortest code length required to encode the data into a binary sequence in a prefix coding method. This is due to the MDL principle [2], which asserts that the best way to model data is to use a model that compresses both the data and the model itself. The MDL principle has been justified in a number of respects such as consistency, rate of convergence (see e.g., [25]) in scenarios involving information theory and statistics. In our setting, we employ Gaussian kernel density estimation as a nonparametric model of the dataset. We then calculate the amount of information in terms of the NML code length of data associated with the class of Gaussian kernel density.

The main process for defining KC can be expressed as follows:

1. We consider a subset of the dataset with constraints on the surrounding data where a certain or higher data density is observed. We derive a parameter $D$ defined as the range of the subset.
2. We calculate the amount of information for the subset with NML code length.
3. We observe the change in the amount of information when $D$ is changed, and then express the complexity of the distribution in terms of the density of the dataset.
2.3. Kernel Complexity

We propose an index that defines the complexity of a distribution in terms of its density. The index is illustrated in Figure 2b. We hypothesize that KC is relatively small when data points are densely distributed (solid blue line). In contrast, KC is relatively large when data points are sparsely distributed (solid orange line). In Figure 2b, the amount of information (denoted by I) increases with D. When the amount of information is biased for the data points in a specific dense area, I is initially considered to increase significantly as D increases, after which the change becomes gradual (see the blue line in Figure 2b). In comparison, when the amount of information varies throughout the dataset, I is considered to gradually increase as D increases (see the orange line in Figure 2b). In this way, this index can be understood as expressing the bias of the amount of information possessed by each data point. By using the Gini coefficient, which is used to express the extent to which data is biased in economics etc., this index can be formulated as follows:

$$KC = 1 - \frac{\int_D I(D) \, dD - \frac{1}{2} \Delta L \cdot \Delta D - L_0 \cdot \Delta D}{\frac{1}{2} \Delta L \cdot \Delta D}$$

where $I(D)$ is a function that defines the amount of information with $D$.

As the integral in equation (1) is difficult to calculate using actual data, we approximate it as follows:

$$\int_D I(D) \, dD \approx \sum_{\ell_p \leq D_{\text{max}}} (\ell_p - \ell_{p-1}) I(\ell_p),$$

where $\ell_p$ is the $p$-th numerical value with $D$ at equal intervals.
2.4. NML Code Length associated with Kernel Density Estimation

In this section, we describe the NML code length, which we use as the definition of the amount of information. Calculation of the NML code length associated with kernel density estimation is not easy and it was therefore necessary to propose a method for this purpose. First, we describe the NML code length as the code length of a dataset in Section 2.4.1. In Section 2.4.2, we derive a method for calculating the normalization term of the NML code length proposed in [26]. In Section 2.4.3, we propose a new method for calculating the NML code length associated with the kernel density.

2.4.1. Normalized Maximum Likelihood

The NML code length is the optimal code length based on Shtarkov’s min–max criterion [24]. We first define the optimal model that satisfies the minimum NML code length. Let an observed data sequence be \( x^n = (x_1, \ldots, x_n) \), where \( x_i = (x_{i1}, \ldots, x_{im})^\top (i = 1, \ldots, n) \). We use a model class \( \mathcal{P} = \{ p(X^n; \theta) : \theta \in \Theta \} \). The NML code length for the data sequence \( x^n \) is defined as follows:

\[
L_{\text{NML}}(x^n) \overset{\text{def}}{=} - \log p(x^n; \hat{\theta}(x^n)) + \log C(M),
\]

where \( C(M) \) is the normalization term and \( \hat{\theta}(x^n) \) is the maximum likelihood estimator calculated as follows:

\[
\hat{\theta}(x^n) \overset{\text{def}}{=} \arg \max_\theta p(x^n; \theta).
\]

2.4.2. Calculation of the Normalization Term

Here we describe a method for calculating the normalization term of the NML code length, which is generally difficult to calculate directly. Thus, we use the indirect method proposed by Rissanen to perform this calculation [26]. In this method, when the maximum likelihood estimator of the distribution parameter \( \theta \) is a sufficient statistic, it can be calculated analytically by the following calculation method.

First, when the maximum likelihood estimator of the distribution parameter \( \theta \) is a sufficient statistic, the distribution \( p \) can be decomposed as follows:

\[
p(y^n; \theta) \, dy^n = f(z|\hat{\theta}) \cdot g(\hat{\theta}; \theta) \, dz \, d\hat{\theta}
\]

where we assume that a mapping \( q \) exists and we can write \( (\hat{\theta}, z) = q(y^n) \), \( f \) is a conditional probability density function, and \( g \) is a probability density function of \( \hat{\theta} \) where \( \theta \) is a parameter. This equality allows us to calculate the normalization term as follows:

\[
C(M) \overset{\text{def}}{=} \int p(y^n; \hat{\theta}(y^n)) \, dy^n = \int \int f(z|\hat{\theta}) \cdot g(\hat{\theta}; \theta) \, dz \, d\hat{\theta} = \int g(\hat{\theta}; \theta) \, d\hat{\theta}.
\]

We use this method to calculate the NML code length associated with kernel density estimation. For a detailed discussion, refer to the book published by Rissanen [26].

2.4.3. NML Code Length associated with Kernel Density Estimation

We next propose a method for calculating the normalization term associated with kernel density estimation. Let an observed data sequence be \( x^n = (x_1, \ldots, x_n) \), \( x_i \in \mathbb{R}^m \). We use the NML code
length and kernel density estimation as a probability density distribution for a given dataset as follows:

\[ f(x; h) = \frac{1}{n} \sum_{j=1}^{n} K(x - x_j; h), \]

where the function \( K(\cdot) \) is a kernel function. In the following discussion, we use the Gaussian kernel \( K(x; h) = 1/(2\pi h^2)^{m/2} \exp\{-||x||^2/2h^2\} \). Aiming to capture the structural changes, we consider calculating the likelihood using only the data points under the following conditions:

\[ A_i = \left\{ j \mid ||x_i - x_j||^2 \leq (1 + \gamma)D \right\} (i = 1, \cdots, n), \]

\[ B = \left\{ i \mid \frac{1}{N(A_i)} \sum_{j \in A_i} ||x_i - x_j||^2 \leq D \right\}, \]

where \( \gamma > 0 \) is a parameter. Under these conditions, we can derive the following theorem.

**Theorem 1.** The NML codelength for the subprobability distribution associated with kernel density is calculated as follows:

\[
\frac{nm}{2} \log \left\{ \sum_{i \in B} \frac{1}{N(A_i)} \sum_{j \in A_i} ||x_i - x_j||^2 \right\} - \sum_{i \in B} \left( \frac{m}{m + 4} \log n - m \log \epsilon \right) + nm \log \left( \frac{n m \pi m}{e} \right) + \log \log \left( \frac{2 \pi D \cdot n \pi m}{m \epsilon^2} \right) + \frac{nm}{2} \log(\pi) - \log \Gamma \left( \frac{nm}{2} \right). \tag{4}
\]

Hereinafter, this NML code length is expressed as \( L_{K-NML}(x^o; \gamma, D) \).

The proof has two parts. The first is the subprobability distribution we introduce with the kernel density estimation. The second is a method we propose for calculating the NML for the subprobability distribution associated with the kernel density estimation in Section 2.4.2. Details of the calculation thereof are presented in Appendix A.

### 2.5. Kernel Complexity with the NML Code Length associated with the Kernel Density Estimate

We define \( KC \) by using the NML code length associated with the kernel density estimation, which is calculated in Section 2.4.3. The definition of \( KC \) based on the NML code length is as follows:

\[
KC_{K-NML} \overset{\text{def}}{=} 1 - \int_D L_{K-NML}(x^o; \gamma, D) \, dD - \frac{1}{2} \Delta L \cdot \Delta D - L_0 \cdot \Delta D \tag{5}
\]

Let us discuss the theoretical nature of the NML code length \( L_{K-NML}(x^o; \gamma, D) \). This NML code length contains the hyperparameters \( \gamma \) and \( D \), and we evaluate the theoretical properties by tracking the change in the NML by \( D \) for fixed \( \gamma \). First, we consider the case in which \( D \) changes by \( \Delta D \). The amount of change in the NML code length can be approximated as follows:

\[
\Delta L = L_{K-NML}(x^o; \gamma, D + \Delta D) - L_{K-NML}(x^o; \gamma, D) \approx \frac{nm}{2} \cdot \frac{1}{1 - \delta} \cdot \frac{N(B(D + \Delta D))}{N(B(D))}, \tag{6}
\]

where \( N(B(D)) \) is the number of \( B \) sets in equation (3), \( \Delta B(D) \) is the increment of set \( B \) when \( D \) changes by \( \Delta D \). Additionally, we define a parameter \( \delta, 0 < \delta < 1 \). Details of the calculation are presented in Appendix B.
Equation (6) is proportional to the rate of increase in $B$ subject to the likelihood calculation. As the value of $\Delta L$ indicates the extent to which the data points subject to the code length increase as $D$ increases, the width of the increase in the amount of information can be regarded as the amount of information of the newly added data points.

2.6. Nature of KC

We calculated the value of KC for several generated datasets to evaluate the nature of KC. Specifically, we evaluated the behavior of the values of KC with respect to the number of sets (clusters in a parametric model) in the dataset and the behavior of the values of KC with respect to the extent of the dataset.

2.6.1. Aggregated Dataset

As a representative of a mixed dataset, we generated an artificial dataset, which we aggregated into several chunks we refer to as clusters. We used the same artificial dataset and performed the aggregation a few times to obtain a different number of clusters, and experimented with the characteristics of the KC values for each of the datasets we produced in this way. The different aggregations we generated are shown in Figure 3. Using this dataset, we calculated the value of KC, which is plotted in Figure 4 as a function of the number of clusters. This result indicates that KC increases as the number of clusters increases. Thus, from the viewpoint of data aggregation, KC is considered to capture the characteristics of the structure of the distribution. In addition to these results, we also experimented with the behavior of KC by changing the dimensions of the dataset. The results are shown in Figure 5a and Figure 5b. These results show that KC increases as the number of clusters increases, and demonstrate the ability of KC to grasp the structural information of data aggregation.

2.6.2. Circular Dataset

As nonparametric data cannot be expressed as a simple block, we investigated the characteristics of KC of a dataset of which the data are aggregated into circular chunks (clusters). As before, we varied the number of clusters, and investigated the extent to which the KC value depends on the number of clusters. The dataset we generated is visualized in Figure 6. Using this dataset, the calculated values of KC are plotted in Figure 7 as a function of the number of circles (i.e., clusters). These results show that KC increases as the number of clusters increases; hence, KC is considered to capture the characteristics of the structure of the distribution in terms of the number of clusters.

Figure 3. Dataset aggregated into several chunks, which we refer to as clusters.

Figure 4. Value of KC as a function of the number of data chunks (clusters) in the dataset.
Based on the above-mentioned findings, we proceeded to apply KC to further investigate its ability to detect structural changes in a dataset. A dataset that undergoes structural change first exhibits minor movement, after which a large change in the structure becomes visible. Because KC is an index for evaluating the complexity of the distribution of a dataset, it can be considered as an indicator of increasing complexity during substantial structural change. The algorithm we developed to compute the KC index for structural change detection is presented in Algorithm 1. For simplicity, we express the dataset as $X_t \in \mathbb{R}^{n \times m}$ and express $KC_{K-NML}(X_t)$ as $KC_t$.

4. Experimental Results

We empirically demonstrate the usefulness of the algorithm using both artificial and practical datasets.
Algorithm 1 Algorithm for structural change detection.

Calculate KC:
Calculate KC score defined by equation (5) as follows:

$$KC_{\text{K-NML}}(X_t) = 1 - \int D L_{\text{K-NML}}(X_t; \gamma, D) dD - \frac{1}{2} \Delta L \cdot \Delta D - L_0 \cdot \Delta D.$$  

Detection of changes:
Detect changes with following conditions:

$$a(t) = \begin{cases} 
1 & \text{if } |KC_t - E[KC_t^{-1}]| > \eta \cdot \text{Std}[KC_t^{-1}], \\
0 & \text{otherwise}. 
\end{cases}$$

We can detect changes in the case where $$a(t) = 1.$$

4.1. Artificial Dataset for Change Detection

4.1.1. Circular Dataset

We generated an artificial dataset distributed on the circumferences of several circles and considered the case in which the number of circles gradually changes over time. The parameters of this dataset are defined as follows:

$$\begin{align*}
\# \text{of circles} &= K, r = (r_1, r_2) & \text{if } 1 \leq t \leq \tau_1, \\
\# \text{of circles} &= K', r = (r_1, r_2, u(t)) & \text{if } \tau_1 + 1 \leq t \leq \tau_2, \\
\# \text{of circles} &= K', r = (r_1, r_2, r_3) & \text{if } \tau_2 + 1 \leq t \leq T,
\end{align*}$$

where $$u(t) = \frac{(\tau_2 - t)r_2 + (t - \tau_1)r_3}{\tau_2 - \tau_1}.$$  

where the parameter $$r$$ denotes the radius of each circle. An example of the generated dataset is shown in Figure 8.

Figure 8. Dataset aggregated into circular data chunks over a period of time.

Figure 9. Calculated value of KC of the dataset aggregated into different circular chunks over time.

We found the points of change by using the index in Algorithm 1. The results are shown in Figure 9, which shows that KC is able to detect a transition in the structure of the data over time and this can be interpreted as an increase in the complexity of the dataset.
In addition to the above qualitative interpretation, we calculated the benefit, delay, FAR, and AUC scores to evaluate the extent to which the algorithm could detect changes. The benefit, delay, FAR, and (area under the curve) AUC scores, respectively, are defined as follows:

\[
\text{benefit} = \begin{cases} 
1 - (\hat{t} - t^*) / U & \text{if } t^* \leq \hat{t} \leq t^* + U, \\
0 & \text{otherwise}; 
\end{cases} 
\]  

(8)

\[
\text{delay} = \begin{cases} 
\hat{t} - t^* & \text{if } \hat{t} \in \text{Transition period}, \\
\text{None} & \text{otherwise}; 
\end{cases} 
\]  

(9)

\[
\text{FAR} = \frac{\#\{\hat{t} \notin \text{Transition period}\}}{\#\{\hat{t} \notin \text{Transition period}\} + 1}; 
\]  

(10)

\[
\text{AUC} = \text{Area under the curve created by plotting the benefit against the FPR.} 
\]  

(11)

where \( \hat{t} \) is the first point at which the algorithm detects a change in the transition period or the point at which the algorithm detects a change in any other period. In this study, the AUC is calculated in relation to the benefit. Using these evaluation scores, we evaluated the proposed algorithm in comparison with the density ratio estimation (DRE) algorithm [19], SDMS algorithm [16], SE algorithm [27], tracking the best expert (TBE) algorithm [12], and the entropy-based method (abbreviated as entropy), which is described in Algorithm 2. When using the DRE algorithm, we process the two-dimensional data \( X_t \in \mathbb{R}^{n \times m} \) at each time as one-dimensional data \( X'_t \in \mathbb{R}^{nm} \) and use them as the input for the DRE algorithm. Even though this is not a perfect fit for the model of the DRE algorithm, it was added to the comparison as a model capable of detecting nonparametric change. The point at which the data complexity begins to rise is defined as the starting point of change in the model, and the proposed method was used to conduct a quantitative comparison experiment using the value of KC.

**Algorithm 2** Entropy algorithm for detecting changes.

**Calculate the density:**
Calculate the density distribution as follows:

\[
f(x; \hat{h}) = \frac{1}{n} \sum_{i=1}^{n} K(x - x_i; \hat{h}), 
\]

where we calculate the estimator \( \hat{h} \) on the basis of the method proposed by Scott [28], which is the default setting in the scipy.stats.gaussian_kde class (see [29]).

**Calculate the entropy:**
Calculate the entropy as follows:

\[
\text{Entropy} = -\int f(y; \hat{h}(x^n)) \log f(y; \hat{h}(x^n)) \, dy. 
\]

**Detection of changes:**
Detect the change points with following conditions:

\[
a(t) = \begin{cases} 
1 & \text{if } |\text{Entropy}_t - E[\text{Entropy}^{t-1}]| > \eta \cdot \text{Std}[\text{Entropy}^{t-1}], \\
0 & \text{if otherwise.} 
\end{cases} 
\]

The results are listed in Table 1. We generated 10 different datasets with the same model and detected the points of change for each dataset. The values provided in the table are the average values of the benefit, delay, and FAR scores. These results show that the proposed algorithm (KC) can detect the points of change to a certain extent in terms of the benefit and delay scores. Although the FAR score of the proposed algorithm was slightly larger than that of the entropy method, this difference is
insignificant. Because the SDMS, SE, and TBE algorithms are based on a parametric Gaussian mixture model (GMM), it is not possible to capture the changes in a circular distribution of data. The DRE algorithm assumes that the data at each time consist of only a single scalar value; thus, it is difficult to detect changes in this problem setting.

The main parameters of the generated data and algorithm are as follows.

- **Dataset parameters:**
  - We generated a circular dataset with $K = 2, K' = 3$ clusters by using equation (7). The radii of the circles were $r = (10, 6, 3)$. A chunk of data starts its transformation at time $t = 50 (= \tau_1 + 1)$ and finishes forming a new chunk of data (a circle with radius $r = 3$) at time $t = 100 (= \tau_2 + 1)$.
  - Each data point contains noise that follows a normal distribution with the standard deviation $\sigma = 0.3$.

- **Algorithm parameters:**
  - We used Algorithm 1 to calculate the index to determine the points of change. The detection parameter is $\eta = 3$, and the maximum value of $D$ is 100. In the evaluation, we defined the parameter $U = 50$ to calculate the evaluation scores and the length of the transition period as 50.

<table>
<thead>
<tr>
<th>Method</th>
<th>Benefit</th>
<th>Delay</th>
<th>FAR</th>
</tr>
</thead>
<tbody>
<tr>
<td>KC</td>
<td>0.632</td>
<td>18.4</td>
<td>0.028</td>
</tr>
<tr>
<td>DRE</td>
<td>0.000</td>
<td>50.0</td>
<td>0.010</td>
</tr>
<tr>
<td>SDMS</td>
<td>0.000</td>
<td>50.0</td>
<td>0.000</td>
</tr>
<tr>
<td>SE</td>
<td>0.000</td>
<td>50.0</td>
<td>0.000</td>
</tr>
<tr>
<td>TBE</td>
<td>0.000</td>
<td>50.0</td>
<td>0.000</td>
</tr>
<tr>
<td>entropy</td>
<td>0.344</td>
<td>32.8</td>
<td>0.007</td>
</tr>
</tbody>
</table>

Table 1. Benefit, delay, and FAR scores for the algorithms (circle dataset).

In addition to these results, we evaluated the experimental results by changing the noise level (from $\sigma = 0.1$ to $\sigma = 0.5$), focusing on the AUC scores. We calculated the AUC scores by changing the detection parameter $\eta$ (from $\eta = 0.5$ to $\eta = 3.0$) in Algorithm 1. We generated aggregations with different patterns of the circular dataset with the following parameters:

1. The radii of the circles were $r = (10, 6, 3)$. The new data chunk is a circle with the radius $r_3 = 3$. The results are summarized in Table 2.
2. The radii of the circles were $r = (10, 3, 6)$. The new data chunk is a circle with the radius $r_3 = 6$. The results are summarized in Table 3.
3. The radii of the circles were $r = (3, 6, 10)$. The new data chunk is a circle with the radius $r_3 = 10$. The results are summarized in Table 4.

These results are similar to those in Table 1. On the basis of all the results, it is concluded that KC is able to detect changes stably, regardless of the difference in the generation model and the value of $\sigma$. The entropy method could detect changes with the next highest score. However, as described in Section 4.2, the entropy method requires computational time of an exponential order for the given dimensions, thus it may be difficult to apply in practice.
Table 2. AUC scores for the algorithms (circular dataset with $r = (10, 6, 3)$).

<table>
<thead>
<tr>
<th>Method</th>
<th>$\sigma = 0.1$</th>
<th>$\sigma = 0.2$</th>
<th>$\sigma = 0.3$</th>
<th>$\sigma = 0.4$</th>
<th>$\sigma = 0.5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>KC</td>
<td>0.946 ± 0.022</td>
<td>0.934 ± 0.026</td>
<td>0.918 ± 0.030</td>
<td>0.919 ± 0.033</td>
<td>0.934 ± 0.023</td>
</tr>
<tr>
<td>DRE</td>
<td>0.495 ± 0.000</td>
<td>0.495 ± 0.000</td>
<td>0.495 ± 0.000</td>
<td>0.495 ± 0.000</td>
<td>0.495 ± 0.000</td>
</tr>
<tr>
<td>SDMS</td>
<td>0.500 ± 0.000</td>
<td>0.500 ± 0.000</td>
<td>0.500 ± 0.000</td>
<td>0.500 ± 0.000</td>
<td>0.500 ± 0.000</td>
</tr>
<tr>
<td>SE</td>
<td>0.500 ± 0.000</td>
<td>0.500 ± 0.000</td>
<td>0.500 ± 0.000</td>
<td>0.500 ± 0.000</td>
<td>0.500 ± 0.000</td>
</tr>
<tr>
<td>TBE</td>
<td>0.500 ± 0.000</td>
<td>0.500 ± 0.000</td>
<td>0.500 ± 0.000</td>
<td>0.500 ± 0.000</td>
<td>0.500 ± 0.000</td>
</tr>
<tr>
<td>entropy</td>
<td>0.930 ± 0.041</td>
<td>0.928 ± 0.050</td>
<td>0.923 ± 0.056</td>
<td>0.908 ± 0.061</td>
<td>0.919 ± 0.052</td>
</tr>
</tbody>
</table>

Table 3. AUC scores for the algorithms (circular dataset with $r = (10, 3, 6)$).

<table>
<thead>
<tr>
<th>Method</th>
<th>$\sigma = 0.1$</th>
<th>$\sigma = 0.2$</th>
<th>$\sigma = 0.3$</th>
<th>$\sigma = 0.4$</th>
<th>$\sigma = 0.5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>KC</td>
<td>0.962 ± 0.016</td>
<td>0.965 ± 0.015</td>
<td>0.967 ± 0.017</td>
<td>0.964 ± 0.018</td>
<td>0.968 ± 0.016</td>
</tr>
<tr>
<td>DRE</td>
<td>0.495 ± 0.000</td>
<td>0.495 ± 0.000</td>
<td>0.495 ± 0.000</td>
<td>0.495 ± 0.000</td>
<td>0.495 ± 0.000</td>
</tr>
<tr>
<td>SDMS</td>
<td>0.500 ± 0.000</td>
<td>0.500 ± 0.000</td>
<td>0.500 ± 0.000</td>
<td>0.500 ± 0.000</td>
<td>0.500 ± 0.000</td>
</tr>
<tr>
<td>SE</td>
<td>0.500 ± 0.000</td>
<td>0.500 ± 0.000</td>
<td>0.500 ± 0.000</td>
<td>0.500 ± 0.000</td>
<td>0.500 ± 0.000</td>
</tr>
<tr>
<td>TBE</td>
<td>0.500 ± 0.000</td>
<td>0.500 ± 0.000</td>
<td>0.500 ± 0.000</td>
<td>0.500 ± 0.000</td>
<td>0.500 ± 0.000</td>
</tr>
<tr>
<td>entropy</td>
<td>0.946 ± 0.032</td>
<td>0.946 ± 0.030</td>
<td>0.944 ± 0.031</td>
<td>0.939 ± 0.031</td>
<td>0.936 ± 0.028</td>
</tr>
</tbody>
</table>

Table 4. AUC scores for the algorithms (circular dataset with $r = (3, 6, 10)$).

<table>
<thead>
<tr>
<th>Method</th>
<th>$\sigma = 0.1$</th>
<th>$\sigma = 0.2$</th>
<th>$\sigma = 0.3$</th>
<th>$\sigma = 0.4$</th>
<th>$\sigma = 0.5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>KC</td>
<td>0.976 ± 0.004</td>
<td>0.977 ± 0.007</td>
<td>0.975 ± 0.009</td>
<td>0.966 ± 0.009</td>
<td>0.966 ± 0.009</td>
</tr>
<tr>
<td>DRE</td>
<td>0.495 ± 0.000</td>
<td>0.495 ± 0.000</td>
<td>0.495 ± 0.000</td>
<td>0.495 ± 0.000</td>
<td>0.495 ± 0.000</td>
</tr>
<tr>
<td>SDMS</td>
<td>0.500 ± 0.000</td>
<td>0.500 ± 0.000</td>
<td>0.500 ± 0.000</td>
<td>0.500 ± 0.000</td>
<td>0.500 ± 0.000</td>
</tr>
<tr>
<td>SE</td>
<td>0.500 ± 0.000</td>
<td>0.500 ± 0.000</td>
<td>0.500 ± 0.000</td>
<td>0.500 ± 0.000</td>
<td>0.500 ± 0.000</td>
</tr>
<tr>
<td>TBE</td>
<td>0.500 ± 0.000</td>
<td>0.500 ± 0.000</td>
<td>0.500 ± 0.000</td>
<td>0.500 ± 0.000</td>
<td>0.500 ± 0.000</td>
</tr>
<tr>
<td>entropy</td>
<td>0.973 ± 0.011</td>
<td>0.965 ± 0.017</td>
<td>0.960 ± 0.015</td>
<td>0.956 ± 0.027</td>
<td>0.950 ± 0.016</td>
</tr>
</tbody>
</table>

4.1.2. Gamma-Distributed Dataset

We generated an artificial dataset that follows the gamma mixture model and considered the case in which the number of mixtures gradually changed over time. The parameters of this dataset are defined as follows:

$$
\begin{align*}
\text{# of Gamma dist.} &= K, k = (k_1, k_2) & \text{if } 1 \leq t \leq \tau_1, \\
\text{# of Gamma dist.} &= K', k = (k_1, k_2, u(t)) & \text{if } \tau_1 + 1 \leq t \leq \tau_2, \\
\text{# of Gamma dist.} &= K, k = (k_1, k_2, k_3) & \text{if } \tau_2 + 1 \leq t \leq T, \\
\text{where } u(t) &= \frac{(t_2 - t)k_3 + (t - \tau_1)k_2}{t_2 - \tau_1},
\end{align*}
$$

where the parameter $k$ denotes the shape of the gamma distribution. An example of the generated dataset is shown in Figure 10, which shows that data points that follow the gamma distribution with $k = 1$ are gradually generated over time.

We used Algorithm 1 to calculate the index to determine the points of change. The results are shown in Figure 11, which shows that KC gradually changes during the transition period and that it detected the changes at the beginning of the transition period.
In addition, we evaluated the experimental results by changing the scale parameter (from $\theta = 0.4$ to $\theta = 2.0$), by focusing on the AUC defined in equation 11. We generated the dataset with the gamma distribution with different data patterns, using the following parameters:

1. The shape parameters of the gamma distribution were $k = (10, 5, 1)$. The new data chunk is a cluster with the shape parameter $k_3 = 1$. The results are summarized in Table 5.
2. The shape parameters of the gamma distribution were $k = (10, 1, 5)$. The new data chunk is a cluster with the shape parameter $k_3 = 5$. The results are summarized in Table 6.
3. The shape parameters of the gamma distribution are $k = (1, 5, 10)$. The new data chunk is a cluster with the shape parameter $k_3 = 10$. The results are summarized in Table 7.

These results indicate that the value of KC and the entropy can be used to detect changes stably and effectively. The DRE and SE produced relatively good results, confirming that the gamma-distributed dataset is a model that easily enables both nonparametric and parametric changes to be detected.

It should be noted that the value of KC and the AUC scores for KC vary depending on the scale parameter. This is probably because a constant value is used in this experiment. This suggests that the integration range of $D$ should be appropriately selected according to the data spread; this is left for future study. In addition, the time required to calculate the entropy is of the exponential order with respect to the data dimension; hence, in practice, the calculation is limited to two dimensions. Therefore, this experiment was conducted using only two-dimensional data, and the behavior of KC when processing high-dimensional data is left for future study.

<table>
<thead>
<tr>
<th>Method</th>
<th>$\theta = 0.4$</th>
<th>$\theta = 0.8$</th>
<th>$\theta = 1.2$</th>
<th>$\theta = 1.6$</th>
<th>$\theta = 2.0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>KC</td>
<td>0.911±0.006</td>
<td>0.878±0.021</td>
<td>0.958±0.006</td>
<td>0.954±0.006</td>
<td>0.942±0.023</td>
</tr>
<tr>
<td>DRE</td>
<td>0.805±0.228</td>
<td>0.805±0.228</td>
<td>0.805±0.228</td>
<td>0.805±0.228</td>
<td>0.805±0.228</td>
</tr>
<tr>
<td>SDMS</td>
<td>0.510±0.046</td>
<td>0.510±0.046</td>
<td>0.510±0.046</td>
<td>0.510±0.046</td>
<td>0.510±0.046</td>
</tr>
<tr>
<td>SE</td>
<td>0.647±0.066</td>
<td>0.661±0.033</td>
<td>0.649±0.064</td>
<td>0.662±0.034</td>
<td>0.649±0.064</td>
</tr>
<tr>
<td>TBE</td>
<td>0.500±0.000</td>
<td>0.500±0.000</td>
<td>0.500±0.000</td>
<td>0.500±0.000</td>
<td>0.500±0.000</td>
</tr>
<tr>
<td>entropy</td>
<td><strong>0.942±0.005</strong></td>
<td><strong>0.939±0.007</strong></td>
<td><strong>0.940±0.006</strong></td>
<td><strong>0.940±0.006</strong></td>
<td><strong>0.939±0.007</strong></td>
</tr>
</tbody>
</table>
Table 6. AUC scores for the algorithms (gamma dataset with shape parameters $k = (10, 1, 5)$).

<table>
<thead>
<tr>
<th>Method</th>
<th>$\theta = 0.4$</th>
<th>$\theta = 0.8$</th>
<th>$\theta = 1.2$</th>
<th>$\theta = 1.6$</th>
<th>$\theta = 2.0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>KC</td>
<td>0.965±0.016</td>
<td>0.900±0.005</td>
<td>0.939±0.031</td>
<td>0.977±0.006</td>
<td>0.878±0.036</td>
</tr>
<tr>
<td>DRE</td>
<td>0.858±0.133</td>
<td>0.858±0.133</td>
<td>0.858±0.133</td>
<td>0.858±0.133</td>
<td>0.858±0.133</td>
</tr>
<tr>
<td>SDMS</td>
<td>0.500±0.000</td>
<td>0.500±0.000</td>
<td>0.500±0.000</td>
<td>0.500±0.000</td>
<td>0.500±0.000</td>
</tr>
<tr>
<td>SE</td>
<td>0.849±0.003</td>
<td>0.849±0.003</td>
<td>0.849±0.004</td>
<td>0.848±0.003</td>
<td>0.848±0.003</td>
</tr>
<tr>
<td>TBE</td>
<td>0.500±0.000</td>
<td>0.500±0.000</td>
<td>0.500±0.000</td>
<td>0.500±0.000</td>
<td>0.500±0.000</td>
</tr>
<tr>
<td>entropy</td>
<td>0.917±0.021</td>
<td>0.915±0.023</td>
<td>0.905±0.025</td>
<td>0.905±0.027</td>
<td>0.904±0.027</td>
</tr>
</tbody>
</table>

Table 7. AUC scores for the algorithms (gamma dataset with shape parameters $k = (1, 5, 10)$).

<table>
<thead>
<tr>
<th>Method</th>
<th>$\theta = 0.4$</th>
<th>$\theta = 0.8$</th>
<th>$\theta = 1.2$</th>
<th>$\theta = 1.6$</th>
<th>$\theta = 2.0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>KC</td>
<td>0.834±0.023</td>
<td>0.893±0.005</td>
<td>0.945±0.009</td>
<td>0.973±0.008</td>
<td>0.969±0.009</td>
</tr>
<tr>
<td>DRE</td>
<td>0.805±0.228</td>
<td>0.805±0.228</td>
<td>0.805±0.228</td>
<td>0.805±0.228</td>
<td>0.805±0.228</td>
</tr>
<tr>
<td>SDMS</td>
<td>0.510±0.046</td>
<td>0.510±0.046</td>
<td>0.510±0.046</td>
<td>0.510±0.046</td>
<td>0.510±0.046</td>
</tr>
<tr>
<td>SE</td>
<td>0.676±0.030</td>
<td>0.676±0.029</td>
<td>0.676±0.029</td>
<td>0.676±0.029</td>
<td>0.676±0.029</td>
</tr>
<tr>
<td>TBE</td>
<td>0.500±0.000</td>
<td>0.500±0.000</td>
<td>0.500±0.000</td>
<td>0.500±0.000</td>
<td>0.500±0.000</td>
</tr>
<tr>
<td>entropy</td>
<td>0.964±0.016</td>
<td>0.966±0.015</td>
<td>0.967±0.013</td>
<td>0.969±0.011</td>
<td>0.969±0.011</td>
</tr>
</tbody>
</table>

4.1.3. Cross Dataset

We generated an artificial dataset distributed along straight lines for several chunks of data points and considered the case in which the number of chunks gradually changed over time. The parameters of this dataset are defined as follows:

\[
\text{# of lines} = K \ (1 \leq t \leq T), \\
a = (a_1, u(t)) \ (1 \leq t \leq T), \\
u(t) = \frac{(T - t + 1) \cdot a_1 + (t - 1) \cdot a_2}{T}.
\]

We generated an artificial dataset with $K = 2$, $a_1 = -0.95$, and $a_2 = 0.95$, where the parameter $a$ denotes the slope of a straight line. An example of the generated dataset is shown in Figure 12, which shows that one of the straight lines is gradually rotating over time.

We used the index in Algorithm 1 to detect the points of change. The results are shown in Figure 13, which indicates that KC is gradually decreasing. This is because the density of the distribution at the center of the distribution becomes relatively larger owing to the gradual movement of one of the straight lines, with an accompanying decrease in the complexity.

In addition, we evaluated the experimental results by changing the noise level (from $\sigma = 0.5$ to $\sigma = 2.5$), by focusing on AUC defined in equation 11.

Table 8. AUC scores for the algorithms (cross dataset).

<table>
<thead>
<tr>
<th>Method</th>
<th>$\sigma = 0.5$</th>
<th>$\sigma = 1.0$</th>
<th>$\sigma = 1.5$</th>
<th>$\sigma = 2.0$</th>
<th>$\sigma = 2.5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>KC</td>
<td>0.908±0.007</td>
<td>0.912±0.005</td>
<td>0.846±0.026</td>
<td>0.915±0.012</td>
<td>0.914±0.013</td>
</tr>
<tr>
<td>DRE</td>
<td>0.495±0.000</td>
<td>0.495±0.000</td>
<td>0.495±0.000</td>
<td>0.495±0.000</td>
<td>0.495±0.000</td>
</tr>
<tr>
<td>SDMS</td>
<td>0.613±0.018</td>
<td>0.495±0.005</td>
<td>0.500±0.000</td>
<td>0.500±0.000</td>
<td>0.500±0.000</td>
</tr>
<tr>
<td>SE</td>
<td>0.615±0.029</td>
<td>0.500±0.022</td>
<td>0.500±0.000</td>
<td>0.500±0.000</td>
<td>0.500±0.000</td>
</tr>
<tr>
<td>TBE</td>
<td>0.582±0.019</td>
<td>0.500±0.000</td>
<td>0.500±0.000</td>
<td>0.500±0.000</td>
<td>0.500±0.000</td>
</tr>
<tr>
<td>entropy</td>
<td>0.782±0.000</td>
<td>0.500±0.000</td>
<td>0.877±0.006</td>
<td>0.871±0.001</td>
<td>0.808±0.015</td>
</tr>
</tbody>
</table>
Considering the overall results, KC is able to detect changes stably, regardless of the difference in the value of $\sigma_t$. For small values of $\sigma_t$, the methods based on parametric models (the SDMS, SE, and TBE algorithms) were sometimes able to detect the points of change, probably because the dataset is relatively closely approximates a GMM.

### 4.1.4. Gaussian Mixture Model

We generated an artificial dataset distributed using a GMM, of which the parameters are defined as follows:

\[
\begin{align*}
K^* = 2, \quad \mu = (\mu_1, \mu_2) & \quad \text{if } 1 \leq t \leq t_1, \\
K^* = 3, \quad \mu = (\mu_1, \mu_2, u(t)) & \quad \text{if } t_1 + 1 \leq t \leq t_2, \\
K^* = 3, \quad \mu = (\mu_1, \mu_2, \mu_3) & \quad \text{if } t_2 + 1 \leq t \leq T,
\end{align*}
\tag{13}
\]

where $u(t) = \frac{(t_2 - t)\mu_2 + (t - t_1)\mu_3}{t_2 - t_1}$.

A chunk of data starts undergoing transformation at time $t = 50$ and finishes forming a new chunk of data at time $t = 100$. An example of the generated dataset is shown in Figure 14.

We used the index in Algorithm 1 to detect the change points. The detection parameter is $\eta = 3$ in Figure 15. During the transition period, KC gradually increases at first, then gradually decreases after reaching its peak, before stabilizing in a certain range. Points of change are indicated by a gradual increase in the value of KC.
We also generated datasets by changing the variance of the Gaussian model from $\sigma^2 = 2.0$ to $\sigma^2 = 10.0$. The resultant AUC scores (see equation 11) are listed in Table 9. KC was able to indicate points of change earlier than the other methods assuming a GMM (SDMS, SE, and TBE algorithm) because it captures changes in the density distribution. In models that assume a GMM (SDMS, SE, and TBE algorithm), the number of clusters changes to a certain degree at a time when a chunk of the dataset undergoes transformation. In contrast, KC can be considered to indicate change in the complexity from the point at which a chunk of the dataset starts to gradually transform. Because of these characteristics, the nonparametric KC could detect change at the earliest time.

Next, we observed the behavior of the benefit and FAR scores when calculating the AUC for different values of the sensitivity parameter. The results are shown in Figure 16, which indicates that KC yields high benefit values, but the corresponding FAR values are also relatively high. This is attributed to KC being more sensitive to change than the other methods. The results in the figure show that each algorithm becomes less sensitive to change as the change sensitivity parameter becomes larger (the change is judged more severely). For SDMS and TBE, which contain no sensitivity parameters, the scores of benefit and FAR were constant. The SE algorithm would be expected to enable change to be detected outside the transition period by reducing the change sensitivity parameter. The values of benefit could be considered to be low because the subsequent time is not detected as a change.

Observing the curve plotting benefit against FAR as shown in Figure 17, the AUC score of KC was higher because the advantages of benefit exceed the disadvantages of FAR.

Table 9. AUC scores for the algorithms (GMM dataset).

<table>
<thead>
<tr>
<th>Method</th>
<th>$\sigma^2 = 2.0$</th>
<th>$\sigma^2 = 4.0$</th>
<th>$\sigma^2 = 6.0$</th>
<th>$\sigma^2 = 8.0$</th>
<th>$\sigma^2 = 10.0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>KC</td>
<td>0.943 ± 0.028</td>
<td>0.965 ± 0.014</td>
<td>0.951 ± 0.018</td>
<td>0.924 ± 0.047</td>
<td>0.930 ± 0.037</td>
</tr>
<tr>
<td>DRE</td>
<td>0.495 ± 0.002</td>
<td>0.495 ± 0.002</td>
<td>0.493 ± 0.006</td>
<td>0.493 ± 0.008</td>
<td>0.492 ± 0.009</td>
</tr>
<tr>
<td>SDMS</td>
<td>0.667 ± 0.187</td>
<td>0.656 ± 0.136</td>
<td>0.731 ± 0.184</td>
<td>0.520 ± 0.065</td>
<td>0.567 ± 0.121</td>
</tr>
<tr>
<td>SE</td>
<td>0.813 ± 0.095</td>
<td>0.784 ± 0.165</td>
<td>0.787 ± 0.157</td>
<td>0.810 ± 0.132</td>
<td>0.714 ± 0.170</td>
</tr>
<tr>
<td>TBE</td>
<td>0.636 ± 0.167</td>
<td>0.589 ± 0.147</td>
<td>0.619 ± 0.135</td>
<td>0.500 ± 0.000</td>
<td>0.489 ± 0.010</td>
</tr>
<tr>
<td>entropy</td>
<td>0.584 ± 0.178</td>
<td>0.922 ± 0.031</td>
<td>0.921 ± 0.034</td>
<td>0.913 ± 0.030</td>
<td>0.914 ± 0.037</td>
</tr>
</tbody>
</table>

Figure 16. Evaluation of KC vs. the other algorithms by plotting benefit and FAR against the change sensitivity parameter.

Figure 17. Evaluation of KC vs. the other algorithms by plotting the benefit against FAR.

4.2. Calculation Time

We evaluated the calculation time of KC and the entropy used above with a number of different dimensions. We used the dataset with the same settings as in Section 2.6.1. To complete the calculation of the entropy in realistic time, the mesh in the calculation was set to rough. The results
are shown in Figure 18, which shows that the time required to compute KC is independent of the dimension. However, the time required to calculate the entropy increases exponentially as the dimension increases.

![Figure 18. Time required by the two methods to perform the calculation for different dimensions.](image)

### 4.3. Real Datasets

#### 4.3.1. Sensor Dataset (Household Electricity Consumption)

We tested our method by using a dataset comprising household electricity consumption measurements provided by Dua and Graff [30]. This dataset contains three categories of electricity consumption data: 1. by the kitchen and laundry room, 2. by an electric water heater, 3. by the air conditioner. The data were acquired every other minute from December 17, 2006 to December 10, 2010. Using this dataset, we defined the features $X_t = (x_{1t}, \cdots, x_{nt})$ and $x_i = (x_{i1}, x_{i2}, x_{i3})$. Each $x_i$ is

![Figure 19. KC at each time (household).](image)

![Figure 20. Average consumption of the first feature (in the kitchen and laundry room) for each week.](image)
the value of the total consumption in an hour for each of the three categories, and each $X_t$ represents the consumption for one week (the number of datasets at each time was $n = 168$ or less).

The results obtained with the proposed method enabled us to observe changes in the data sequence at a specific time. The value of KC is shown in Figure 19 as a function of time. We could detect large changes at $t = 87, 88$ (August 17–23, 2008 and August 24–30, 2008). As seen in Figure 20, the use of electricity in the kitchen and laundry room on a normal day is always greater than zero; however, the consumption is zero for the week in which the change is detected. This is because we were able to observe some lifestyle anomalies (or changes) by calculating the value of KC.

4.3.2. Marketing Dataset (Beer Purchasing Behavior)

![Figure 21. KC at each time (beer).](image)

We tested our method using a dataset containing data of beer purchases that was provided by Hakuhodo, Inc. and M-CUBE, Inc. QPR. The dataset is a record of customers’ beer purchasing behavior and includes brands from six manufacturers (denoted A–F here). We captured the changes in the requirements of high-value customers by using simulation and analyzed the data using the top 50% of customers in terms of purchasing volume. The dataset at each instant represents the amount purchased for the four categories of beer in the last two weeks, where the target period was from November 1 to January 31 of the following year.

The results obtained with the proposed algorithm to calculate the value of KC to detect the points of change are shown in Figure 21. The proposed algorithm detected points of significantly large change on December 1, 3, and 31 and on January 1, 13, 14, and 15 ($t = 17, 19, 47, 48, 60, 61, 62$). The detection of change in customers’ purchasing behavior in the last two weeks of November and December reflects the year-end demand, and the complexity of the structure decreases because the beer consumption stabilizes after the end of the year. These results confirm the ability of the proposed algorithm to effectively capture real market changes.
5. Conclusion

This paper proposed a method to calculate the value of KC to define new structural information for data with a nonparametric distribution and proposed a method to detect its change over time. This index is defined by measuring the density of data in terms of information bias and is based on the Gini index. We use the NML code length based on the MDL principle as a criterion to express information. We showed that this index, KC, is a value that characterizes the number of data chunks. Furthermore, we proposed an algorithm to detect the change in KC when the data are given as time series. By using this algorithm, we provide a framework for the detection of changes based on KC.

However, KC has some limitations. First, since the parameters that define KC are not determined with theory, it is a future research issue to improve the reliability of KC by establishing parameter determination theory. Second, the assumed data structure is a structure with data chunks, and the usefulness of KC has only been shown in experiments focusing on the number of data chunks. For this reason, we consider that application to other data structures or proposal of new methods is a future research issue.

The usefulness of the proposed method was experimentally demonstrated using both artificial and real datasets. For the artificial datasets, the proposed algorithm could detect the change points in terms of the benefit, delay, FAR, and AUC scores for specific kinds of datasets. Further, we showed the effectiveness of our method for analyzing a dataset containing household electricity consumption data. Specifically, our algorithm automatically detected changes at times when the electricity consumption in the kitchen and laundry room was likely to change. In addition, we also analyzed a dataset containing data relating to customers’ beer purchasing behavior over time. The purchasing behavior significantly changed over the last part of the year and after the year ended, and the proposed algorithm effectively captured these changes.

The ability of the proposed method to capture the complexity in a data structure that can be defined by the density has expanded the possibility of searching for new hidden value. In future, we aim to extend our work to other kernel functions, and to calculate exact values rather than the upper bound in the form of the NML. With respect to the KC index itself, analysis of the theoretical nature of this indicator (e.g., its properties by using the Gini coefficient) remains as an extension of current research. This may enable us to obtain appropriate values for the parameters (the features of the KC index depend on the parameters to some extent). In terms of applications, we consider adding more qualitative interpretations of actual data in combination with other methods. In addition, considering the application of our method, real data are rarely neatly arranged; thus, extending KC to an index that can handle missing data is a very important issue.


Funding: This work was partially supported by JST KAKENHI 19H05802 and JST-AIP JPMJCR19U4.

Acknowledgments: The marketing dataset was provided by Hakuhodo Inc. and M-CUBE, Inc. QPR.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A Calculation of the NML Code Length with Subprobability Distribution associated with Kernel Density Estimation

First, we consider a bandwidth estimator with the kernel density function. We derive the log-likelihood as follows:

$$
\log f(x^n; h) = - \sum_{i \in B} \log \left( N(A_i)(2\pi h^2)^{\frac{m}{2}} \right) + \sum_{i \in B} \log \sum_{j \in A_i} \exp \left\{ - \frac{1}{2h^2} ||x_i - x_j||^2 \right\},
$$
where \( N(A_i) \) and \( N(B) \) are the number of \( A_i \) and \( B \) sets, respectively. Then, using the inequality \( \log \left( \frac{1}{n} \sum_{i=1}^{n} x_i \right) \geq \frac{1}{2} \sum_{i=1}^{n} \log x_i \), the lower bound of this log-likelihood can be calculated as follows:

\[
\log f(x^n; h) = -\sum_{i \in B} \log \left( (2\pi h^2)^{\frac{p}{2}} \right) + \sum_{i \in B} \log \left( \frac{1}{N(A_i)} \sum_{j \in A_i} \frac{1}{2h^2} ||x_i - x_j||^2 \right) \\
\geq -\sum_{i \in B} \log \left( (2\pi h^2)^{\frac{p}{2}} \right) - \sum_{i \in B} \frac{1}{N(A_i)} \sum_{j \in A_i} \frac{1}{2h^2} ||x_i - x_j||^2.
\]

(A1)

The bandwidth \( h \) of a kernel density was optimized to be \( h = O(n^{-1/(m+4)}) \) in the past work (e.g., refer to [28]) so that the generalization error for the maximum likelihood estimator for \( h \) is minimal. When considering the NML code length associated with the kernel, we set a constraint of \( h \geq e \cdot n^{-1/(m+4)} / \sqrt{2\pi} \) using a positive constant \( e \). Under this condition, equation (A1) can be lower-bounded as follows:

Equation (A1)

\[
\begin{align*}
\hat{L}(x^n; h) & = -N(B) \log \left( (2\pi h^2)^{\frac{p}{2}} \right) - \sum_{i \in B} \frac{1}{N(A_i)} \sum_{j \in A_i} \frac{1}{2h^2} ||x_i - x_j||^2 \\
& = -N(B) \log \left( (2\pi h^2 \cdot n^{\frac{p}{2m+4}} \cdot \frac{1}{e^2})^{\frac{p}{2}} \right) - \sum_{i \in B} \left\{ \frac{1}{N(A_i)} \sum_{j \in A_i} \frac{1}{2h^2} ||x_i - x_j||^2 - \frac{m}{m+4} \log n + m \log e \right\} \\
& \geq -n \log \left( (2\pi h^2 \cdot n^{\frac{p}{2m+4}} \cdot \frac{1}{e^2})^{\frac{p}{2}} \right) - \sum_{i \in B} \left\{ \frac{1}{N(A_i)} \sum_{j \in A_i} \frac{1}{2h^2} ||x_i - x_j||^2 - \frac{m}{m+4} \log n + m \log e \right\} \\
& =: \hat{L}(x^n; h).
\end{align*}
\]

(A2)

Then, we calculate the estimator \( \tilde{h} \) such that \( \hat{L}(x^n; h) \) is maximized as follows:

\[
\tilde{h}(x^n) = \frac{1}{nm} \sum_{i \in B} \frac{1}{N(A_i)} \sum_{j \in A_i} ||x_i - x_j||^2.
\]

We define the distribution \( \tilde{f} \) as follows:

\[
\begin{align*}
\tilde{f}(x^n; h) & \overset{\text{def}}{=} \exp \left\{ \hat{L}(x^n; h) \right\} \\
& = \frac{1}{(2\pi h^2 \cdot n^{\frac{p}{2m+4}} \cdot \frac{1}{e^2})^{\frac{p}{2}}} \exp \left\{ -\frac{1}{2h^2} \sum_{i \in B} \frac{1}{N(A_i)} \sum_{j \in A_i} ||x_i - x_j||^2 \right\} \times \exp \left\{ \sum_{i \in B} \left( \frac{m}{m+4} \log n - m \log e \right) \right\}.
\end{align*}
\]

Using this equation and equation (A2), we find that the distribution \( \tilde{f} \) is a subprobability distribution by the following formula:

\[
\int \tilde{f}(x^n; h) dx^n \leq \int f(x^n; h) dx^n = 1.
\]

Then, the NML distribution of \( \tilde{f} \) can be calculated as follows:

\[
\begin{align*}
\tilde{f}_{\text{NML}}(x^n) & \overset{\text{def}}{=} \frac{\tilde{f}(x^n; \tilde{h}(x^n))}{C}, \\
C & \overset{\text{def}}{=} \int \tilde{f}(y^n; \tilde{h}(y^n)) dy^n,
\end{align*}
\]
where $C$ is a normalization term of the NML distribution. In general, the normalization term cannot be calculated in a straightforward manner; instead, we calculate this term using the method described in Sec. 2.4.2. We can decompose $\tilde{f}(x^n; h)$ as follows:

$$\tilde{f}(x^n; h) \, dx^n = \tilde{f}(z; \tilde{h}) \cdot g(\tilde{h}; h) \, dz \, \tilde{h},$$

where the function $g$ is the gamma distribution with the shape parameter $k = nm/2$ and scale parameter $\theta = 2h^2/nm$:

$$g(\tilde{h}; h) = \frac{2\tilde{h}}{\Gamma \left(\frac{nm}{2}\right)} \left(\frac{2h^2}{nm}\right)^{\frac{nm}{2}} \exp \left\{ - \frac{\tilde{h}^2}{2h^2/nm} \right\}.$$

Then, we can define

$$g(\tilde{h}) \overset{\text{def}}{=} g(\tilde{h}; h) = \frac{2 \exp \left( - \frac{nm}{2} \right)}{\Gamma \left(\frac{nm}{2}\right)} \frac{1}{\tilde{h}}.$$

Then, we can calculate the normalization term $C$ for integrating with respect to $\tilde{h}$ as follows:

$$C = \int_Y \tilde{f}(y^n; \tilde{h}(y^n)) dy^n
= \int_{c^{-1/\sqrt{2\pi} \sqrt{D/m}}}^{\sqrt{D/m}} g(\tilde{h}) d\tilde{h}
= \frac{2 \exp \left( - \frac{nm}{2} \right)}{\Gamma \left(\frac{nm}{2}\right)} \left(\frac{2}{nm}\right)^{\frac{nm}{2}} \log \left( \sqrt{\frac{2\pi D \cdot n^{\frac{2}{nm+4}}}{mc^2}} \right),$$

where we define the range of $\tilde{h}$ as $Y = [1/\sqrt{2\pi} \cdot \sqrt{D/m}]$. The upper bound on $\tilde{h}$ is calculated as follows:

$$\tilde{h}(x^n) = \sqrt{\frac{1}{nm} \sum_{i \in B} \frac{1}{N(A_i)} \sum_{j \in A_i} ||x_i - x_j||^2}
\leq \sqrt{\frac{1}{nm} \sum_{i \in B} D}
\leq \sqrt{\frac{D}{m}}.$$
Finally, we can calculate the NML code length using the subprobability distribution \( \hat{f} \) as follows:

\[
- \log \hat{f}_{\text{NML}}(x^n) = - \log \left\{ \exp\left(\frac{-nm}{2}\right) \cdot \exp \left\{ \sum_{i \in B} \left( \frac{m}{m+4} \log n - m \log \epsilon \right) \right\} \frac{1}{2\pi n \pi^{\frac{n}{2}} \cdot \frac{1}{2}} \right\} \\
+ \log \left\{ \frac{2 \exp\left(\frac{-nm}{4}\right)}{\Gamma\left(\frac{nm}{2}\right)} \cdot \log \left( \frac{2 \pi D \cdot n \pi^{\frac{n}{2}}}{me^2} \right) \right\} \\
= \frac{nm}{2} \log \left\{ \sum_{i \in B} \frac{1}{N(A_i)} \sum_{j \in A_i} ||x_i - x_j||^2 \right\} - \sum_{i \in B} \left( \frac{m}{m+4} \log n - m \log \epsilon \right) \\
+ nm \log \left( \frac{n \pi^{\frac{n}{2}}}{\epsilon} \right) + \log \log \left( \frac{2 \pi D \cdot n \pi^{\frac{n}{2}}}{me^2} \right) + \frac{nm}{2} \log(\pi) - \log \Gamma\left(\frac{nm}{2}\right).
\]

### Appendix B Approximation of \( \Delta D \)

We describe the approximation of \( \Delta L \) discussed in Section 2.3. First, we can approximate \( \Delta L \) as follows:

\[
\Delta L = L_{K-\text{NML}}(x^n; \gamma, D + \Delta D) - L_{K-\text{NML}}(x^n; \gamma, D) \\
\approx \frac{nm}{2} \log \left\{ \sum_{i \in B(D)} \frac{1}{N(A_i(D))} \sum_{j \in A_i(D)} ||x_i - x_j||^2 + \sum_{i \in \Delta B(D)} D \right\} \\
- \frac{nm}{2} \log \left\{ \sum_{i \in B(D)} \frac{1}{N(A_i(D))} \sum_{j \in A_i(D)} ||x_j - x_j||^2 \right\}, \tag{A3}
\]

where \( N(A_i(D)) \) and \( N(B(D)) \) are the number of \( A_i \) and \( B \) sets in equation (3), respectively, and \( \Delta B(D) \) is the increment of \( B \) when changing \( D \) by \( \Delta D \). As we assume that \( \Delta D \) is sufficiently small, we can naturally assume that

\[
\sum_{i \in \Delta B(D)} D \ll \sum_{i \in B(D)} \frac{1}{N(A_i(D))} \sum_{j \in A_i(D)} ||x_i - x_j||^2.
\]

Using this assumption, we approximate equation (A3) as follows:

\[
\text{Equation (A3)} \approx \frac{nm}{2} \frac{1}{\sum_{i \in B(D)} \frac{1}{N(A_i(D))} \sum_{j \in A_i(D)} ||x_i - x_j||^2} \sum_{i \in \Delta B(D)} D. \tag{A4}
\]

Using a parameter \( \delta, 0 < \delta < 1 \), we can rewrite this equation as follows:

\[
\sum_{i \in B(D)} \frac{1}{N(A_i(D))} \sum_{j \in A_i(D)} ||x_i - x_j||^2 = N(B(D)) \cdot (1 - \delta) D. \tag{A5}
\]

Using this formula, we can calculate equation (A4) as follows:

\[
\text{Equation (A4)} = \frac{nm}{2} \frac{\sum_{i \in \Delta B(D)} D}{N(B(D)) \cdot (1 - \delta) D} = \frac{nm}{2} \cdot \frac{1}{1 - \delta} \frac{N(\Delta B(D))}{N(B(D))}. \tag{A6}
\]


18. Gama, J.; Žliobaité, I.; Bifet, A.; Pechenizkiy, M.; Bouchachia, A. A survey on concept drift adaptation. *ACM computing surveys (CSUR)* 2014, 46, 44.


