



Detecting signs of model change with continuous model selection based on descriptive dimensionality

Kenji Yamanishi¹ · So Hirai¹

Accepted: 11 June 2023 / Published online: 24 August 2023
© The Author(s) 2023

Abstract

We address the issue of detecting changes of models that lie behind a data stream. The model refers to an integer-valued structural information such as the number of free parameters in a parametric model. Specifically we are concerned with the problem of how we can detect signs of model changes earlier than they are actualized. To this end, we employ *continuous model selection* on the basis of the notion of *descriptive dimensionality* (Ddim). It is a real-valued model dimensionality, which is designed for quantifying the model dimensionality in the model transition period. Continuous model selection is to determine the real-valued model dimensionality in terms of Ddim from a given data. We propose a novel methodology for detecting signs of model changes by tracking the rise-up/descent of Ddim in a data stream. We apply this methodology to detecting signs of changes of the number of clusters in a Gaussian mixture model and those of the order in an auto regression model. With synthetic and real data sets, we empirically demonstrate its effectiveness by showing that it is able to visualize well how rapidly model dimensionality moves in the transition period and to raise early warning signals of model changes earlier than they are detected with existing methods.

Keywords Model change detection · Change sign detection · Minimum description length principle · Model selection · Continuous model selection · Descriptive dimension · Clustering

1 Introduction

1.1 Motivation

This paper is concerned with the issue of detecting changes of a model that lies behind a data stream. The *model* refers to the discrete structural information such as the number of free parameters in the mechanism for generating the data. We consider the situation where a model changes over time. Under this environment, it is important to detect the model changes as accurately as possible. This is because the model changes may correspond to important events. For example, it is reported in [14] that when customers' behaviors are modeled using a Gaussian mixture model, the change of

the number of mixture components corresponds to the emergence or disappearance of a cluster of customers' behaviors. In this case a model change implies a change of the market trend. For another example, it is reported in [35] that when the syslog behaviors are modeled using a mixture of hidden Markov models, the change of the number of mixture components may correspond to a system failure.

The issue of model change detection has extensively been explored. This paper is rather concerned with the issue of detecting *signs* or *early warning signals* of model changes. Why is it important to detect such signs? One reason is that if they were detected earlier than the changes themselves, we could predict the changes before they were actualized. The other reason is that if they were detected after the change themselves, we could analyze the cause of the changes in a retrospective way.

A model, say, the number of parameters, is an integer-valued index, in general. Therefore, it appears that the model change abruptly occurs. However, it is reasonable to suppose that some intrinsic change, which we call *latent change*, gradually occurs at the back of the model change. Then we may define a *sign* of the model change as the starting point of the

✉ Kenji Yamanishi
yamanishi@g.ecc.u-tokyo.ac.jp
So Hirai
So.Hirai.16@gmail.com

¹ School of Information Science and Technology,
The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo
113-8656, Japan

latent change. Therefore, if we properly define a real-valued index to quantify the model dimensionality in the transition period, we can understand how rapidly the latent change is going on and we can detect signs of model changes by tracking the rise-up/descent of the index (Fig. 1).

The key idea of this paper is to employ the notion of *descriptive dimensionality* (Ddim) for the quantification of a model in the transition period. Ddim is a real-valued index, which quantifies the model dimensionality for the case where a number of models are mixed. We thereby establish a methodology of *continuous model selection*. It is to determine the optimal real-valued model dimensionality from data on the basis of Ddim. In the transition period of model changes, the mixing structure of models may change over time. Hence, by tracking the rise-up/descent of Ddim, we will be able to track the latent changes behind model changes.

The purpose of this paper is twofold: One is to establish a novel methodology for detecting signs (or early warning signals) of model changes from a data stream. We realize this by using Ddim for the quantification of model dimensionality in its transition period. The theory of Ddim is developed on the basis of the *minimum description length* (MDL) principle [24] in combination with the theory of box counting dimension. The other is to empirically validate the effectiveness of the methodology using synthetic and real data sets.

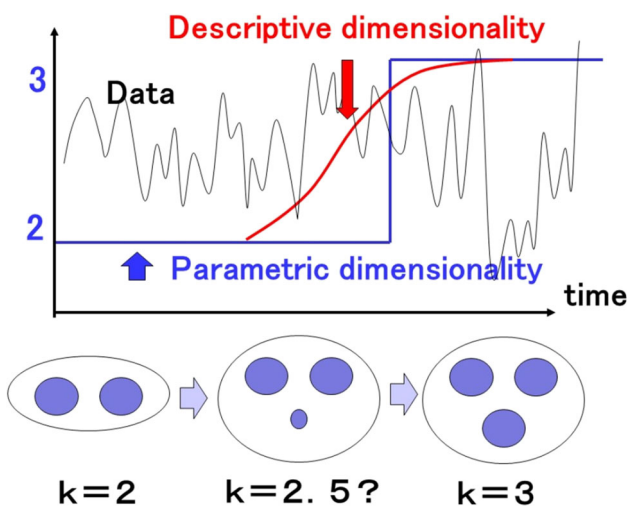


Fig. 1 Transition period of dimensionality change. We consider the situation where the clustering structure changes over time so that the number k of clusters changes from $k = 2$ to $k = 3$. Here k can be thought of as an integer-valued model dimensionality, called the *parametric dimensionality*. If we define a real-valued intrinsic model dimensionality, called the *descriptive dimensionality*, then we can quantify the dimensionality in the transition period. For example, k becomes 2.5 at some time point. By tracking the rise-up/descent of such a real-valued model dimensionality, we are able to detect signs of increase of the number of clusters

We evaluate how early and how reliably it is able to make alarms of signs of model changes.

1.2 Related work

Model change detection has been studied in the scenario of *dynamic model selection* (DMS) developed in [34, 35]. Model change detection is different from the classical continuous parameter change detection. Taking an example of finite mixture models, the former is to detect changes in the number of components, while the latter is to detect those in the real-valued parameters of individual components or mixing parameters. In [34, 35], they proposed the DMS algorithm, which outputs a model sequence of the shortest description length, on the basis of the MDL principle [24]. They demonstrated its effectiveness from the empirical and information-theoretic aspects. The MDL based model change detection has been further theoretically justified in [33]. The problems similar to model change detection have been discussed in the scenarios of switching distributions [8], tracking best experts [13], on-line clustering [27], cluster evolution [21], Bayesian change detection [29], and structure break detection for autoregression model [3]. In all of these previous studies, however, a model change was considered to be an abrupt change of a discrete structure. The transition period of changes has never been analyzed there. In the conventional state-space model, change detection of continuous states is addressed (see e.g.[5]). Then the state itself has not the same meaning as a model which we define in this paper. The number of states is a model which we mean.

Changes that do not occur abruptly but incrementally occur were discussed in the context of detecting incremental changes in concept drift [10], gradual changes [36], volatility shift [17], etc. However, it has never been quantitatively analyzed how rapidly a model changes in the transition period.

Recently, the indices of structural entropy [16] and graph-based entropy [22] have been developed for measuring the uncertainty associated with model changes. Although they can be thought of as early warning signals of model changes, they cannot quantify the intrinsic model dimensionality nor explain how rapidly a model changes in the transition period. Change sign detection method using differential MDL change statistics has been proposed in [37]. However, it is applied to change sign detection for parameters only. We summarize the related work in Table 1 from the viewpoints of abrupt model change detection, model change sign detection, and quantification of model dimensionality.

This paper proposes a methodology for analyzing model transition in terms of real-valued dimensionality. A number of notions of dimensionality have been proposed in the areas of physics and statistics. The metric dimension was proposed by Kolmogorov and Tihomirov [18] to measure the complexity of a given set of points in terms of the notion of covering

Table 1 Comparison of related works on model change detection

methods	abrupt model change detection	model change sign detection	quantification of model dimensionality
dynamic model selection [34, 35], tracking best experts [13], on-line clustering [27], Bayesian change detection [29], cluster evolution [21], structure break detection [3] MDL change statistics [33, 36] concept drift [10], volatility shift [17]	○	×	×
structural entropy [16], graph-based entropy [22]	×	○	×
Ddim based method (this study)	○	○	○

numbers. This was evolved into the notion of the box counting dimension, equivalently, the fractal dimension [20]. It is a real-valued index for quantifying the complexity of a given set. It is also related to the capacity [7]. Vapnik Chervonenkis dimension was proposed to measure the power of representation for a given class of functions [28]. It was also related to the rate of uniform convergence of estimating functions. See [12] for relations between dimensionality and learning. The dimensionality as a power of representation is conventionally integer-valued, but when it changes over time, there is no effective non-integer valued quantification of its transition. The previous notions of dimensionality are summarized in Table 2 from the viewpoints of integer/real-valued, characterization of learning rate, and quantification of model change.

Preliminary versions of this paper appeared in Arxiv [31, 32].

1.3 Significance of this paper

The significance of this paper is summarized as follows: (1) *Proposal of a novel methodology for detecting signs of model changes with continuous model selection.* This paper proposes a novel methodology for detecting signs of model changes. The key idea is to track model transitions with

continuous model selection using the notion of *descriptive dimensionality* (Ddim). It measures the model dimensionality in the case where a number of models with different dimensionalities are mixed.

For example, we employ the Gaussian mixture model (GMM) to consider the situation where the number of mixture components changes over time. We suppose that in the transition period of model change, a number of probabilistic models with various mixture sizes are fused. We give a method for calculating Ddim for this case. The transition period of model change can be visualized by drawing a Ddim graph versus time. Once a Ddim graph is obtained, we can understand how rapidly the model changes over time. We eventually detect signs of model changes by tracking the rise-up/descent of Ddim. This methodology is significantly important in data mining since it helps us predict model changes in earlier stages.

(2) *Empirical demonstration of effectiveness of model change sign detection via Ddim.* We empirically validate how early we are able to detect signs of model changes with continuous model selection, for GMMs and auto-regression (AR) models. With synthetic data sets and real data sets, we illustrate that our method is able to effectively visualize the transition period of model change using Ddim. We further empirically demonstrate that our methodology

Table 2 Comparison of related works on dimensionality

methods	real-valued/integer-valued	characterization of learning rate	quantification of model change
metric dimension [18], VC dimension [28],	integer-valued	○	×
fractal dimension [20], capacity [7]	real-valued	×	×
descriptive dimension (this study)	real-valued	○	○

is able to detect signs of model changes significantly earlier than any existing dynamic model selection algorithms and is comparable to structural entropy in [16]. Through our empirical analysis, we demonstrate that Ddim is an effective index for measuring the model dimensionality in the model transition period.

(3) *Giving theoretical foundations for Ddim.* In this paper, Ddim plays a central role in continuous model selection. We introduce this notion from an information-theoretic view based on the MDL principle [24] (see also [11]). We show that Ddim coincides with the number of free parameters in the case where the model consists of a single parametric class. We also derive Ddim for the case where a number of models with different dimensionalities are mixed. We characterize Ddim by demonstrating that it governs the rate of convergence of the MDL-based learning algorithm. This corresponds to the fact that the metric dimensionality governs the rate of convergence of the empirical risk minimization algorithm in statistical learning theory [12].

The rest of this paper is organized as follows: Section 2 introduces the notion of Ddim. Section 3 gives a methodology for model change sign detection via Ddim. Section 4 shows experimental results. Section 5 characterizes Ddim by relating it to the rate of convergence of the MDL learning algorithm. Section 6 gives conclusion. Source codes and data sets are available at a Github repository [38].

2 Descriptive dimensionality

2.1 NML and parametric complexity

This section introduces the theory of Ddim. This theory is based on the MDL principle (see [24] for the original paper and [25] for the recent advances) from the viewpoint of information theory. We start by introducing a number of fundamental notions of the MDL principle.

Let \mathcal{X} be the data domain where \mathcal{X} is either discrete or continuous. Without loss of generality, we assume that \mathcal{X} is discrete. Let $\mathbf{x} = x_1, \dots, x_n \in \mathcal{X}^n$ be a data sequence of length n . We assume that each x_i is independently generated. $\mathcal{P} = \{p(\mathbf{x})\}$ be a class of probabilistic models where $p(\mathbf{x})$ is a probability mass function or a probability density function. Hereafter, we assume that for any \mathbf{x} , the maximum of $p(\mathbf{x})$ with respect to p exists.

Under the MDL principle, the information of a datum \mathbf{x} is measured in terms of description length, i.e., the codelength required for encoding the datum with a prefix coding method. We may encode \mathbf{x} with help of a class \mathcal{P} of probability distributions. One of the most important methods for calculating the codelength \mathbf{x} using \mathcal{P} is the normalized maximum likelihood (NML) coding [25]. This is defined as the codelength associated with the NML distribution as follows:

Definition 1 We define the *normalized maximum likelihood (NML) distribution* over \mathcal{X}^n with respect to \mathcal{P} by

$$p_{\text{NML}}(\mathbf{x}; \mathcal{P}) \stackrel{\text{def}}{=} \frac{\max_{p \in \mathcal{P}} p(\mathbf{x})}{\sum_{\mathbf{y}} \max_{p \in \mathcal{P}} p(\mathbf{y})}. \tag{1}$$

The *normalized maximum likelihood (NML) codelength* of \mathbf{x} relative to \mathcal{P} , which we denote as $L_{\text{NML}}(\mathbf{x}; \mathcal{P})$, is given as follows:

$$\begin{aligned} L_{\text{NML}}(\mathbf{x}; \mathcal{P}) &\stackrel{\text{def}}{=} -\log p_{\text{NML}}(\mathbf{x}; \mathcal{P}) \\ &= -\log \max_{p \in \mathcal{P}} p(\mathbf{x}) + \log C_n(\mathcal{P}), \end{aligned} \tag{2}$$

where

$$\log C_n(\mathcal{P}) \stackrel{\text{def}}{=} \log \sum_{\mathbf{y}} \max_{p \in \mathcal{P}} p(\mathbf{y}). \tag{3}$$

The first term in (2) is the negative logarithm of maximum likelihood while the second term (3) is the logarithm of the normalization term. The latter is called the *parametric complexity* of \mathcal{P} [25]. This means the information-theoretic complexity for the model class \mathcal{P} relative to the length n of data sequence. The NML codelength can be thought of as an extension of Shannon information $-\log p(\mathbf{x})$ into the case where the true model p is unknown but only \mathcal{P} is known.

In order to understand the meaning of the NML codelength and the parametric complexity, we define the *minimax regret* as follows:

$$R_n(\mathcal{P}) \stackrel{\text{def}}{=} \min_q \max_{\mathbf{x}} \left\{ -\log q(\mathbf{x}) - \min_{p \in \mathcal{P}} (-\log p(\mathbf{x})) \right\},$$

where the minimum is taken over the set of all probability distributions. The minimax regret means the descriptive complexity of the model class, indicating how largely any codelength is deviated from the smallest negative log-likelihood over the model class. Shtarkov [26] proved that the NML distribution (1) is optimal in the sense that it attains the minimum of the minimax regret. In this sense the NML codelength is the optimal codelength for encoding \mathbf{x} for given \mathcal{P} . Then we can immediately see that the minimax regret coincides with the parametric complexity. That is,

$$R_n(\mathcal{P}) = C_n(\mathcal{P}). \tag{4}$$

We next consider how to calculate the parametric complexity. According to [25] (pp:43-44), the parametric complexity can be rewritten using a variable transformation technique as follows:

$$C_n(\mathcal{P}) = \sum_{\mathbf{y}} \max_{p \in \mathcal{P}} p(\mathbf{y}) = \int g(\hat{p}, \hat{p}) d\hat{p}, \tag{5}$$

where $g(\hat{p}, p)$ is defined as

$$g(\hat{p}, p) \stackrel{\text{def}}{=} \sum_{\mathbf{y}: \max_{\tilde{p} \in \mathcal{P}} \tilde{p}(\mathbf{y}) = \hat{p}(\mathbf{y})} p(\mathbf{y}). \tag{6}$$

2.2 Definition of descriptive dimension

Below we give the definition of Ddim from a view of approximation of the parametric complexity, equivalently, the minimax regret (by (4)). The scenario of defining Ddim is as follows: We first count how many points are required to approximate the parametric complexity (5) with quantization. We consider that count as information-theoretic richness of representation for a model class. We then employ that count to define Ddim in a similar manner with the box counting dimension.

We consider to approximate (5) with a finite sum of partial integrals of $g(\hat{p}, \hat{p})$. Let $\bar{\mathcal{P}} = \{p_1, p_2, \dots\} \subset \mathcal{P}$ be a finite subset of \mathcal{P} . Let ϵ be the parameter for defining the diameter of the neighborhood of a given probability distribution. For $\epsilon > 0$, for $p_i \in \bar{\mathcal{P}}$, let $D_\epsilon^n(i) \stackrel{\text{def}}{=} \{p \in \mathcal{P} : d_n(p_i, p) \leq \epsilon^2\}$ where $d_n(p_i, p)$ is the Kullback-Leibler (KL) divergence between p and p_i :

$$d_n(p, p_i) = \frac{1}{n} \sum_{\mathbf{x}} p_i(\mathbf{x}) \log \frac{p_i(\mathbf{x})}{p(\mathbf{x})}.$$

Then we approximate $\bar{C}_n(\bar{\mathcal{P}})$ by

$$\bar{C}_n(\bar{\mathcal{P}}) \stackrel{\text{def}}{=} \sum_i Q_\epsilon(i), \tag{7}$$

where

$$Q_\epsilon(i) \stackrel{\text{def}}{=} \int_{\hat{p} \in D_\epsilon^n(i)} g(\hat{p}, \hat{p}) d\hat{p}. \tag{8}$$

That is, (7) gives an approximation to $C_n(\mathcal{P})$ with a finite sum of integrals of $g(\hat{p}, \hat{p})$ over the ϵ^2 -neighborhood of a point p_i . We define $m_n(\epsilon : \mathcal{P})$ as the smallest number of points $|\bar{\mathcal{P}}|$ with respect to $\bar{\mathcal{P}}$ such that $C_n(\mathcal{P}) \leq \bar{C}_n(\bar{\mathcal{P}})$. More precisely,

$$m_n(\epsilon : \mathcal{P}) \stackrel{\text{def}}{=} \min_{\bar{\mathcal{P}}} |\bar{\mathcal{P}}| \quad \text{subject to } C_n(\mathcal{P}) \leq \bar{C}_n(\bar{\mathcal{P}}). \tag{9}$$

We are now led to the definition of descriptive dimension.

Definition 2 [31] Let \mathcal{P} be a class of probability distributions. We let $m(\epsilon : \mathcal{P})$ be the one obtained by choosing $\epsilon^2 n = O(1)$ in $m_n(\epsilon : \mathcal{P})$ as in (9). We define the *descriptive dimension* (Ddim) of \mathcal{P} by

$$\text{Ddim}(\mathcal{P}) \stackrel{\text{def}}{=} \lim_{\epsilon \rightarrow 0} \frac{\log m(\epsilon : \mathcal{P})}{\log(1/\epsilon)}, \tag{10}$$

when the limit exists.

The definition of Ddim is similar with that of the *box counting dimension* [7, 9, 20]. The main difference between them is how to count the number of points. Ddim is calculated on the basis of the number of points required for approximating the parametric complexity, while the box counting dimension is calculated on the basis of the number of points required for covering a given object with their ϵ -neighborhoods.

Consider the case where \mathcal{P}_k is a k -dimensional parametric class, i.e., $\mathcal{P}_k = \{p(\mathbf{x}; \theta) : \theta \in \Theta_k \subset \mathbb{R}^k\}$, where Θ_k is a k -dimensional real-valued parameter space. Let $p(\mathbf{x}; \theta) = f(\mathbf{x} | \hat{\theta}(\mathbf{x}))g(\hat{\theta}(\mathbf{x}); \theta)$ for the conditional probabilistic mass function $f(\mathbf{x} | \hat{\theta}(\mathbf{x}))$. We then write g according to (6) as follows

$$g(\hat{\theta}, \theta) = \sum_{\mathbf{x}: \arg \max_{\theta'} p(\mathbf{x}; \theta') = \hat{\theta}} p(\mathbf{x}; \theta). \tag{11}$$

Assume that the central limit theorem holds for the maximum likelihood estimator of a parameter vector θ . Then according to [25], we can take a Gaussian density function as (11) asymptotically. That is, for sufficiently large n , (11) can be approximated as:

$$g(\hat{\theta}, \theta) \simeq \left(\frac{n}{2\pi}\right)^{\frac{k}{2}} |I_n(\theta)|^{\frac{1}{2}} e^{-n(\hat{\theta}-\theta)^\top I_n(\theta)(\hat{\theta}-\theta)/2}, \tag{12}$$

where $I_n(\theta) \stackrel{\text{def}}{=} (1/n) E_\theta[-\partial^2 \log p(\mathbf{x}; \theta) / \partial \theta \partial \theta^\top]$ is the Fisher information matrix.

The following theorem shows the basic property of $m_n(\epsilon : \mathcal{P}_k)$ for the parametric case.

Theorem 1 Suppose that $p(\mathbf{x}; \theta) \in \mathcal{P}_k$ is continuously three-times differentiable with respect to θ . Under the assumption of the central limit theorem so that (12) holds, for sufficiently large n , we have

$$\log C_n(\mathcal{P}_k) = \log m_n(1/\sqrt{n} : \mathcal{P}_k) + O(1). \tag{13}$$

The proof is given in Appendix.

It is known [25] (p.53) that under some regularity condition that the central limit theorem holds for the maximum likelihood estimator for θ , the parametric complexity for \mathcal{P}_k is asymptotically expanded as

$$\log C_n(\mathcal{P}_k) = \frac{k}{2} \log \frac{n}{2\pi} + \log \int \sqrt{|I(\theta)|} d\theta + o(1), \tag{14}$$

where $I(\theta)$ is the Fisher information matrix: $I(\theta) \stackrel{\text{def}}{=} \lim_{n \rightarrow \infty} (1/n) \times E_\theta[-\partial^2 \log p(\mathbf{x}; \theta) / \partial \theta \partial \theta^\top]$. Plugging (13) with (14) for $\epsilon^2 n = O(1)$ into (10) yields the following theorem.

Theorem 2 For a k -dimensional parametric class \mathcal{P}_k , under the regularity condition for \mathcal{P}_k as in Theorem 1, we have

$$Ddim(\mathcal{P}_k) = k. \tag{15}$$

Theorem 2 shows that when the model class is a single parametric one, $Ddim$ coincides with the conventional notion of dimensionality (the number of free parameters), which we call the *parametric dimensionality* in the rest of this paper.

$Ddim$ can also be defined even for the case where the model class is not a single parametric class. Hence Theorem 2 implies that $Ddim$ is a natural extension of the parametric dimensionality.

Let us consider *model fusion* where a number of model classes are probabilistically mixed. Let $\mathcal{F} = \{\mathcal{P}_1, \dots, \mathcal{P}_s\}$ be a family of model classes and assume a model class is probabilistically distributed according to $p(\mathcal{P})$ over \mathcal{F} . We denote the model fusion over \mathcal{F} as $\mathcal{F}^\odot = \mathcal{P}_1 \odot \dots \odot \mathcal{P}_s$. We may interpret the resulting distribution over \mathcal{X} as a finite mixture model [19] of a number of model classes with different dimensionalities. Then $Ddim$ of \mathcal{F}^\odot is calculated as

$$\begin{aligned} \lim_{\epsilon \rightarrow 0} \frac{\log E_{\mathcal{P}}[m(\epsilon : \mathcal{P})]}{\log(1/\epsilon)} &\geq \lim_{\epsilon \rightarrow 0} \sum_{i=1}^s p(\mathcal{P}_i) \frac{\log m(\epsilon : \mathcal{P}_i)}{\log(1/\epsilon)} \\ &= \sum_{i=1}^s p(\mathcal{P}_i) Ddim(\mathcal{P}_i), \end{aligned} \tag{16}$$

where we have used Jensen’s inequality to derive the first inequality. We call the lower bound (16) the *pseudo $Ddim$ for model fusion \mathcal{F}^\odot* . In the rest of this paper, we adopt it as $Ddim$ value for model fusion. We write it as $\underline{Ddim}(\mathcal{F}^\odot)$. Model fusion is a reasonable setting when we consider the transition period of model changes. Then $Ddim$ is no longer integer-valued.

3 Model change sign detection

3.1 Continuous model selection for GMMs

This section proposes a methodology for detecting signs of model changes with continuous model selection. We first focus on the case where the model is a *Gaussian mixture model* (GMM). The problem setting is as follows: At each time we obtain a number of unlabeled multi-dimensional examples. By observing such examples sequentially, we obtain a data stream of the examples. At each time, we may conduct clustering of the examples using GMMs. Assuming that the number of components in GMM may change over time, we aim at detecting their changes and the signs of them.

The key idea is to conduct continuous model selection, which is to determine the real-valued model dimensionality

on the basis of $Ddim$. Below we give a scenario of continuous model selection with applications to model change sign detection. Let \mathcal{P}_k be a class of GMMs with k components. We consider the situation where the structure of GMM gradually changes over time (Fig. 2), while the model k may abruptly change. The key observation is that during the model transition period, model fusion occurs where a number of GMMs with different k s are probabilistically mixed according to the posterior probability distribution. Then model dimensionality in the transition period can be calculated as $Ddim$ of model fusion. Thus we can detect signs of model changes by tracking the rise-up/descent of $Ddim$.

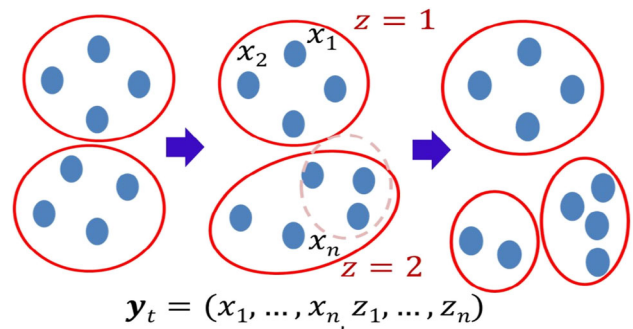
Below let us formalize the above scenario. Let \mathcal{X} be an m -dimensional real-valued domain and let $x \in \mathcal{X}$ be an observed datum. Let $z \in \{1, \dots, k\}$ be a latent variable indicating which component x comes from. Let $\mu_i \in \mathbb{R}^m$, $\Sigma_i \in \mathbb{R}^{m \times m}$ be the mean vector and variance-covariance matrix for the i th component, respectively. Let $\mu = (\mu_1, \dots, \mu_k)$ and $\Sigma = (\Sigma_1, \dots, \Sigma_k)$. Let $\sum_i \pi_i = 1$, $\pi_i \geq 0$ ($i = 1, \dots, k$). Let $\theta = (\mu_i, \Sigma_i, \pi_i)_{|i=1, \dots, k}$. Then a complete variable model of GMM with k -components is given by

$$p(x, z; \theta, k) = p(x | z; \mu, \Sigma) p(z; \pi),$$

where

$$\begin{aligned} p(x | z = i; \mu, \Sigma) &= \frac{1}{(2\pi)^{\frac{m}{2}} \cdot |\Sigma_i|^{\frac{1}{2}}} \exp \left\{ -\frac{1}{2} (x - \mu_i)^\top \Sigma_i^{-1} (x - \mu_i) \right\}, \\ p(z = i; \pi) &= \pi_i \quad (i = 1, \dots, k). \end{aligned} \tag{17}$$

Let $\mathbf{x} = x_1, \dots, x_n$ be a sequence of observed variables of length n . Let z_j denote a latent variable which corresponds to x_j and $\mathbf{z} = z_1, \dots, z_n$. Let $\mathbf{y} = (\mathbf{x}, \mathbf{z})$ be a complete variable.



$$Ddim = 2 \quad 2 < Ddim < 3 \quad Ddim = 3$$

Fig. 2 Continuous model selection. We consider the situation where the number k of clusters in a GMM changes from two to three as time goes by. In the transition period, one cluster gradually collapses into two clusters. $Ddim$ is a continuous variant of the number of clusters, hence $Ddim$ may change continuously in the transition period, taking the value in $(2, 3)$. It represents the gradual change of clustering structure well

Let $\hat{\mu}_i, \hat{\Sigma}_i$ be the maximum likelihood estimators of μ_i, Σ_i ($i = 1, \dots, k$) for given \mathbf{y} . Let $\hat{\pi}_i = n_i/n$ where n_i is the number of occurrences in \mathbf{z} such that $z = i$ ($i = 1, \dots, k$) and $\sum_{i=1}^k n_i = n$. \mathbf{z} may be estimated by sampling from the posterior probability obtained by the EM algorithm. Let $\hat{\theta}(\mathbf{y}) = (\hat{\pi}_i, \hat{\mu}_i, \hat{\Sigma}_i)_{i=1, \dots, k}$.

The NML codelength of \mathbf{y} for a complete variable model of a GMM is given by

$$L_{\text{NML}}(\mathbf{y}; k) = -\log p_{\text{NML}}(\mathbf{y}; k) = -\log p(\mathbf{y}; \hat{\theta}(\mathbf{y}), k) + \log C_n(k), \tag{18}$$

where $C_n(k)$ is a parametric complexity for a GMM. According to [15], an upper bound on $C_n(k)$ is given as follows:

$$C_n(k) \leq \sum_{n_1, \dots, n_k} \frac{n!}{n_1! \dots n_k!} \times \prod_{i=1}^k \left(\frac{n_i}{n}\right)^{n_i} B(m, R, \epsilon) \times \left(\frac{n_i}{2e}\right)^{\frac{mn_i}{2}} \left(\Gamma_m\left(\frac{n_i-1}{2}\right)\right)^{-1}, \tag{19}$$

where

$$B(m, R, \epsilon) \stackrel{\text{def}}{=} \frac{2^{m+1} R^{\frac{m}{2}} \epsilon^{-\frac{m^2}{2}}}{m^{m+1} \cdot \Gamma\left(\frac{m}{2}\right)},$$

where R is a positive constant such that for all $i, \|\hat{\mu}_i\|^2 \leq R$, and ϵ is a positive constant such that ϵ is the lower bound on the smallest eigenvalue of Σ_i for any i . Γ_m is the multivariate Gamma function defined as $\Gamma_m(x) = \pi^{\frac{m(m-1)}{4}} \prod_{j=1}^m \Gamma(x + \frac{1-j}{2})$ and Γ is the Gamma function. We use the bound (19) as the value of $C_n(k)$. It is known [14] that $C_n(k)$ is computable in time $O(n^2k)$.

At each time t , we observe a data sequence: $\mathbf{x}_t = x_1, \dots, x_n \in \mathcal{X}^n$ of length n . We sequentially observe such a datum as show in Fig. 2. Let $\mathbf{x}^T = \mathbf{x}_1, \dots, \mathbf{x}_T$ ($\mathbf{x}_t \in \mathcal{X}^n, t = 1, \dots, T$) be an observed data sequence. The length n may vary over time. We denote the joint sequence of observed variables and latent variables at time t as $\mathbf{y}_t = (\mathbf{x}_t, \mathbf{z}_t)$.

We suppose that a number of GMMs with different k s are fused according to the probability distribution $p(k | \mathbf{y}_t)$ at each time. We define $p(k | \mathbf{y}_t)$ as the annealed posterior probability of k for \mathbf{y}_t :

$$p(k | \mathbf{y}_t) \stackrel{\text{def}}{=} \frac{(p_{\text{NML}}(\mathbf{y}_t; k) p(k|k_{t-1}))^\beta}{\sum_{k'} (p_{\text{NML}}(\mathbf{y}_t; k') p(k'|k_{t-1}))^\beta} \tag{20}$$

$$= \frac{\exp(-\beta L_{\text{NML}}(\mathbf{x}_t, \mathbf{z}_t; k) + \beta \log p(k|k_{t-1}))}{\sum_{k'} \exp(-\beta L_{\text{NML}}(\mathbf{x}_t, \mathbf{z}_t; k') + \beta \log p(k'|k_{t-1}))},$$

where k_{t-1} is the dimensionality estimated at time $t - 1$, and

$$p(k | k_{t-1}) \stackrel{\text{def}}{=} \begin{cases} 1 - \gamma & \text{if } k = k_{t-1} \text{ and } k_{t-1} \neq 1, k_{\text{max}}, \\ 1 - \gamma/2 & \text{if } k = k_{t-1} \text{ and } k_{t-1} = 1, k_{\text{max}}, \\ \gamma/2 & \text{if } k = k_{t-1} \pm 1. \end{cases} \tag{21}$$

γ ($0 < \gamma < 1$) is a parameter, and k_{max} is the maximum value of k . We estimate γ using the MAP estimator with the beta distribution $\text{Beta}(a, b)$ being the prior where the density function of $\text{Beta}(a, b)$ is proportional to $\gamma^{a-1}(1-\gamma)^{b-1}$ for hyper-parameters a and b . The MAP estimator of γ is given as follows:

$$\hat{\gamma} = \frac{N_t + a - 1}{t + a + b - 2},$$

where N_t shows how many times the number of clusters has changed until time $t - 1$. In the experiments to follow, we set $(a, b) = (2, 10)$. β (> 0) is the temperature parameter. In our experiment, we set as β as

$$\beta = 1/\sqrt{n}. \tag{22}$$

This is due to the PAC-Bayesian argument [1] for Gibbs posteriors.

Note that (20) is calculated on the basis of the NML distribution. This is because the probability distribution with unknown parameters should be estimated as the NML distribution since it is the optimal distribution in terms of the minimax regret (see Section 2.1).

By (16), we can calculate Ddim of model fusion of GMMs with various k s at time t as

$$\underline{\text{Ddim}}(\mathcal{F}^\odot)_t = \sum_k p_t(\mathcal{P}_k) \text{Ddim}(\mathcal{P}_k), \tag{23}$$

where $p_t(\mathcal{P}_k)$ is the probability of \mathcal{P}_k at time t , and $p_t(\mathcal{P}_k) = p(k | \mathbf{y}_t)$ in this case. Note that Ddim for GMM with k components is $k(m^2/2 + (5m/2)) - 1 \approx kf(m)$ where $f(m) = m^2/2 + (5m/2)$. However, in order to focus on the mixture size, we divide the true Ddim by $f(m)$ to consider an alternative Ddim of the form of (24).

$$\underline{\text{Ddim}}(\mathcal{F}^\odot)_t \stackrel{\text{def}}{=} \sum_k p(k | \mathbf{y}_t) \text{Ddim}(\mathcal{P}_k) / f(m) \approx \sum_k p(k | \mathbf{y}_t) k. \tag{24}$$

The calculation of real-valued k according to (24) is really continuous model selection.

Suppose that there exists a true parametric dimensionality k^* . Then because of the *consistency* of MDL model estimation ([25], pp:63-69),

$$p(\hat{k} = k^* \mid \mathbf{y}_t) \rightarrow 1$$

for \hat{k} minimizing the NML codelength as n increases, Hence (24) will coincide with k^* with probability 1 as n goes to infinity. This implies that (24) is a natural extension of parametric dimensionality.

3.2 Model change sign detection algorithms

Consider the situation where we sequentially observe a complete variable sequence: $\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_T$. We then obtain a Ddim graph:

$$\{(t, \underline{\text{Ddim}}(\mathcal{F}^\odot)_t) : t = 1, 2, \dots, T\},$$

as in Fig. 1. We can visualize the transition period by drawing the Ddim graph versus time. In this paper we have defined a sign of a model change as the starting point of the latent gradual change associated with it. Thus we can detect signs of model changes by looking at the rise-up/descent of Ddim.

More precisely, we propose the following two methods for raising alarms of model change signs.

1) *Thresholding method* (TH): We raise an alarm if the absolute difference between Ddim and the baseline exceeds a given threshold δ_1 . The baseline is the parametric dimensionality estimated by the *sequential dynamic model selection algorithm* (SDMS) [14], which is a sequential variant of DMS in [34, 35]. It outputs a model $k = \hat{k}$ with the shortest code-length, i.e., for $\lambda > 0$,

$$\hat{k} = \underset{k}{\operatorname{argmin}} \{L_{\text{NML}}(\mathbf{y}_t; k) - \lambda \log p(k \mid k_{t-1})\}, \tag{25}$$

where $L_{\text{NML}}(\mathbf{y}_t; k)$ is calculated as in (18). Letting \hat{k} be the output of SDMS and $\underline{\text{Ddim}}_t$ be Ddim of model fusion at time t , we raise an alarm if

$$\text{TH-Score} \stackrel{\text{def}}{=} |\underline{\text{Ddim}}_t - \hat{k}| > \delta_1. \tag{26}$$

2) *Differential method* (Diff): We raise an alarm if the time difference of Ddim exceeds a given threshold δ_2 . That is, we raise an alarm if

$$\text{Diff-Score} \stackrel{\text{def}}{=} |\underline{\text{Ddim}}_t - \underline{\text{Ddim}}_{t-1}| > \delta_2. \tag{27}$$

The computational complexity of TH and Diff at each time t is governed by that for computing the NML codelength (18). The first term in (18) is computable in time $O(nk)$. The second term in (18) is computable in time $O(n^2k)$ [15], but it

does not depend on data, hence can be calculated for various n and k beforehand. It can be referred when necessary. Hence the computational complexity of TH and Diff at each time is $O(nK)$ where K is an upper bound on k .

3.3 Continuous model selection for AR model

The above methodology can be applied to general classes of finite mixture models other than GMMs. It can also be applied to general classes of parametric probabilistic model classes. We illustrate the case of *auto-regression* (AR) model as an example. This model does not include latent variables.

Let a data sequence $\{x_t\}$, $x_t \in \mathbb{R}$ ($t = 1, 2, \dots, T$) be given. For the modeling of the data sequence, we consider $AR(k)$ (k -th order auto-regression) model of the form:

$$x_t = a_1 x_{t-1} + \dots + a_k x_{t-k} + \epsilon,$$

where $a_i \in \mathbb{R}$ ($i = 1, \dots, k$) are unknown parameters, and ϵ is a random variable following the Gaussian distribution with mean 0 and unknown variance σ^2 . We set $\theta = (a_1, \dots, a_k, \sigma^2)$.

Let $\mathbf{x}_t = x_t, \dots, x_{t-w+1}$ be the t -th session for a window size w . We calculate the NML codelength $L_{\text{NML}}(\mathbf{x}_t; k)$ for \mathbf{x}_t associated with $AR(k)$ in the following sequential manner: Letting $\hat{\theta}$ be the maximum likelihood estimator,

$$L_{\text{NML}}(\mathbf{x}_t; k) = \sum_{j=t-w+1}^t \left(-\log \frac{p(x_j; \hat{\theta}(x_j, x^{j-1}))}{\int p(y_j; \hat{\theta}(y, x^{j-1})) dy} \right).$$

Letting $\mathcal{P}_k = AR(k)$, similarly with (20) and (23), we can calculate Ddim at time t and draw a Ddim graph. We thereby detect model change signs by applying TH and Diff to the graph.

4 Experimental results

4.1 Synthetic data: GMM

4.1.1 Data set

We employ synthetic data sets to evaluate how well we are able to detect signs of model changes using Ddim. We let $n = 1000$ at each time. We generated DataSet 1 according to GMMs so that the number of components changed from $k = 2$ to $k = 3$ as follows:

$$\begin{cases} k = 2, \mu = (\mu_1, \mu_2) & \text{if } 0 \leq t \leq \tau_1, \\ k = 3, \mu = (\mu_1, \mu_2, f_\alpha(t)) & \text{if } \tau_1 + 1 \leq t \leq \tau_2, \\ k = 3, \mu = (\mu_1, \mu_2, \mu_3) & \text{if } \tau_2 + 1 \leq t \leq T, \end{cases}$$

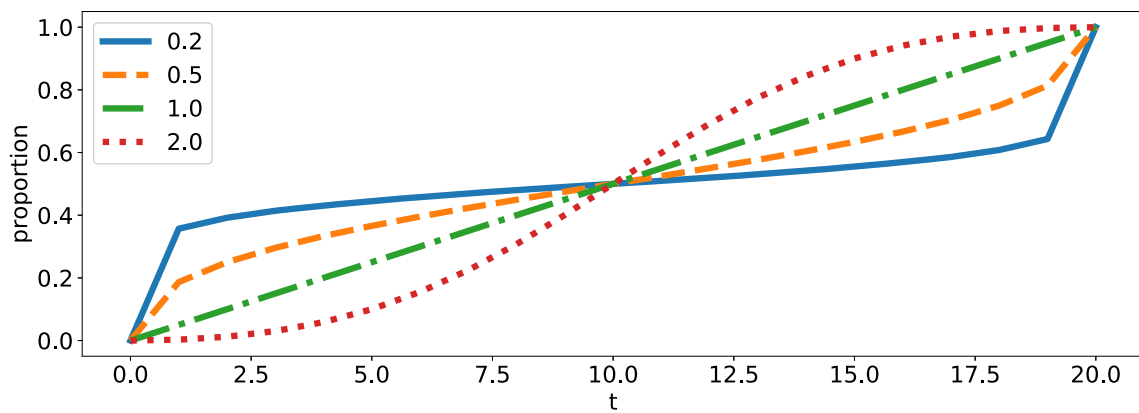


Fig. 3 Graph of proportion of μ_3 in the mean to μ_2 versus time for various α

where letting α be a smoothness parameter,

$$f_\alpha(t) \stackrel{\text{def}}{=} \frac{(\tau_2 - t)^\alpha \mu_2 + (t - \tau_1)^\alpha \mu_3}{(\tau_2 - t)^\alpha + (t - \tau_1)^\alpha} \quad (\alpha > 0). \tag{28}$$

In it, one component collapsed gradually in the transition period from $t = \tau_1 + 1$ to $t = \tau_2$. $f_\alpha(t)$ is the mean value which switches from μ_2 to μ_3 where the speed of change is specified by a parameter α . Figure 3 shows the graph of the proportion of μ_3 in the mean to μ_2 versus time for various α . The change becomes rapid as α approaches to zero. The variance covariance matrix of each component is given by

$$\Sigma = (rAA^\top + (I - r)I) \times \text{var}, \tag{29}$$

where $r = 0.2$, $\text{var} = 3$, and A is a randomly generated $m \times m$ matrix. We set $m = 3$, $\tau_1 = 9$, $\tau_2 = 29$, $T = 39$. It appears that the number of components of a GMM abruptly changed at $t = 20$ since it takes a discrete value. However, in the early stage of $k = 3$, the model is very close to $k = 2$ because the mean values of Gaussian components are very close each other. It may be more natural to recognize the model dimensionality at this stage as a value between $k = 2$ and $k = 3$.

We evaluate how well Ddim tracked the transition period of model change. The temperature parameter β was chosen so that $\beta = 0.0316$ according to (22). Figure 4 shows how Ddim gradually grows as time goes by for various α values. The gray zone shows the transition period when the model changes from $k = 2$ to $k = 3$. The blue line shows the number of components of the GMM estimated by the SDMS algorithm as in (25). The green curve shows the Ddim graph. The red and purple curves show TH-Score and Diff-Score as in (26) and (27), respectively. We show the time points of their alarms of TH and Diff using the same colors. Ddim successfully visualized how rapidly the GMM structure changed in the transition period from $t = 10$ to 29. The true change occurs rapidly for $\alpha = 0.2$, while it occurs slowly

for $\alpha = 1.0$. Ddim was able to successfully track their transition process depending on α . Ddim detected signs earlier than SDMS made an alarm of model change.

We next consider the case where there are multiple change points. We generated DataSet 2 according to GMMs so that the number of components changed from $k = 2$ to $k = 3$, then from $k = 3$ to $k = 4$. as follows:

$$\begin{cases} k = 2, \mu = (\mu_1, \mu_2) & \text{if } 1 \leq t \leq \tau_1, \\ k = 3, \mu = (\mu_1, \mu_2, \frac{(\tau_2 - t)^\alpha \mu_2 + (t - \tau_1)^\alpha \mu_3}{\tau_2 - \tau_1}) & \text{if } \tau_1 + 1 \leq t \leq \tau_2, \\ k = 3, \mu = (\mu_1, \mu_2, \mu_3) & \text{if } \tau_2 \leq t \leq \tau_3, \\ k = 4, \mu = (\mu_1, \mu_2, \mu_3, \frac{(\tau_4 - t)^\alpha \mu_3 + (t - \tau_3)^\alpha \mu_4}{\tau_4 - \tau_3}) & \text{if } \tau_3 + 1 \leq t \leq \tau_4, \\ k = 4, \mu = (\mu_1, \mu_2, \mu_3, \mu_4) & \text{if } \tau_4 + 1 \leq t \leq T. \end{cases}$$

One component collapsed gradually over time from $t = 10$ to $t = 29$ and the other one collapsed from $t = 50$ to $t = 69$. We set $\tau_1 = 9$, $\tau_2 = 29$, $\tau_3 = 49$, $\tau_4 = 69$, and $T = 79$. In the transition periods the parameters varied as with the single change point case.

Figure 5 shows the Ddim graph for $\alpha = 0.5$. The gray zone shows the transition periods when the model changes from $k = 2$ to $k = 3$ and $k = 3$ to $k = 4$. The green curve shows the Ddim graph. The blue line shows the number of components of the GMM estimated by SDMS. The red and purple lines show the times when alarms for signs of model changes are raised using TH and Diff, respectively. The Ddim graph helps us understand well how rapidly the GMM structure gradually changes in the transition periods from $t = 10$ to $t = 29$ and from $t = 50$ to $t = 69$. The Ddim graph helps us understand well how rapidly the model changes in the transition period. TH detected the signs of model changes earlier than SDMS.

4.1.2 Evaluation metrics

Next we quantitatively evaluate how early we were able to detect signs of model changes with Ddim. We measure the performance of any algorithm in terms of benefit. Let \hat{t} be

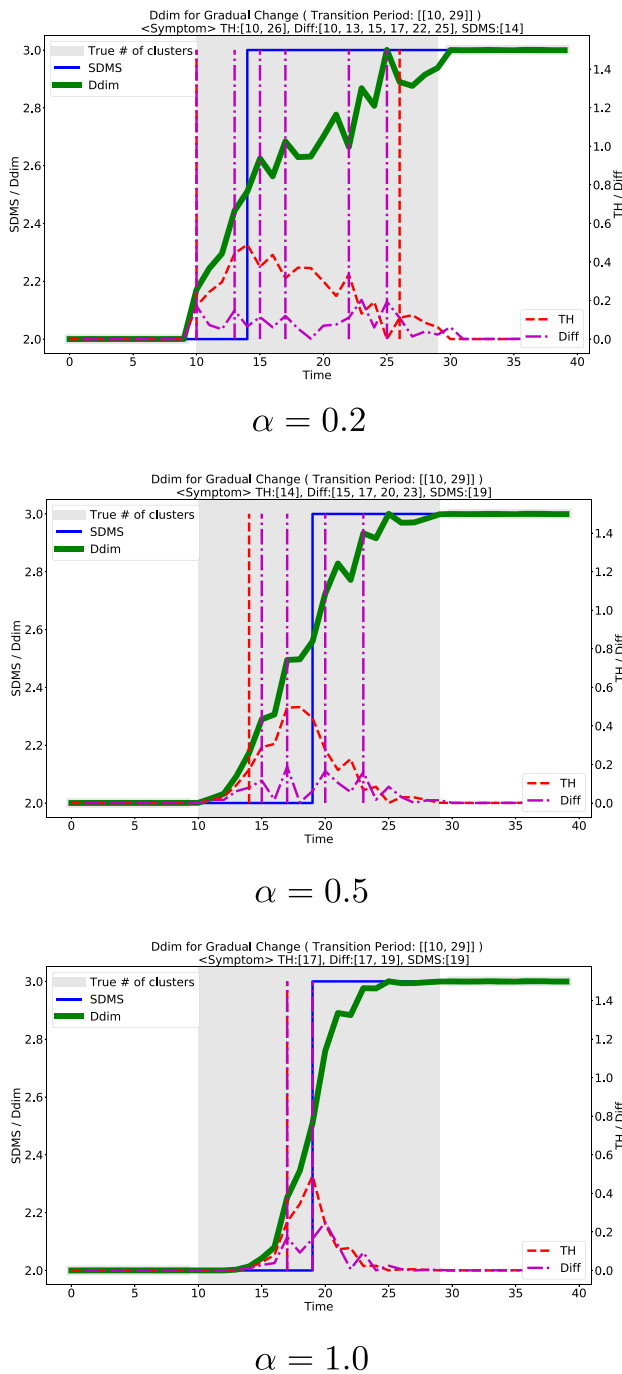


Fig. 4 Ddim graph (transition period: $[\tau_1 = 9, \tau_2 = 29], T = 39$). We see how Ddim gradually grows as time goes by for various α values. The gray zone shows the transition period when the model changes from $k = 2$ to $k = 3$. The blue line shows the number of components of the GMM estimated by the SDMS algorithm as in (25). The green curve shows the Ddim graph. The red and purple curves show TH-Score and Diff-Score as in (26) and (27), respectively. We show the time points of their alarms of TH and Diff using the same colors. We see that Ddim successfully visualized how rapidly the GMM structure changes in the transition period from $t = 10$ to 29 for each α , where the true change occurs rapidly for $\alpha = 0.2$, while it occurs slowly for $\alpha = 1.0$. Furthermore, Ddim detected signs of model change earlier than SDMS made an alarm of model change

the first time when an alarm is made and t^* be the true sign, which we define as *the starting point of model change*. Then *benefit* is defined as

$$\text{benefit} = \begin{cases} 1 - (\hat{t} - t^*)/U & (t^* \leq \hat{t} < t^* + U), \\ 0 & \text{otherwise,} \end{cases} \quad (30)$$

where U is a given parameter. Benefit takes the maximum value 1 when the alarm coincides with the true sign. It decreases linearly as t goes by and becomes zero as \hat{t} exceeds $t^* + U$.

False alarm rate (FAR) is defined as the ratio of the number of alarms outside the transition period over the total number of alarms.

We evaluate any method for model change sign detection algorithm in terms of Area Under Curve (AUC) of Benefit-FAR curve that is obtained by varying the threshold parameter δ such as in (26) and (27). We set $U = 10$ in (30).

4.1.3 Methods for comparison

We consider the following methods for comparison.

1) *The sequential DMS algorithm* (SDMS) [14]: The SDMS algorithm with $\lambda = 1$ outputs the estimated parametric dimensionality as in (25). We raise an alarm when the output of SDMS changes.

2) *Fixed share algorithm* (FS) [13]: We think of each model k as an *expert*, and perform Herbster and Warmuth's *fixed share algorithm*, abbreviated as FS. It was originally designed to make prediction by taking a weighted average over a number of experts, where the weight is calculated as a linear combination of the exponential update weight and the sum of other experts' ones. In it the expert with the largest weight is the best expert, which may change over time. We can think of FS as a model change detection algorithm by tracking the time-varying best expert.

Here is a summary of FS. Let k be the index of the expert. $L_k(\mathbf{z}_{t-1})$ is the loss function for the k th expert for data \mathbf{z}_{t-1} , which is the NML codeword in our setting. $w_{t,k}^u$ and $w_{t,k}^s$ are tentative and final weights for the k th expert at time t . FS conducts the following weight update rule: Letting $\alpha > 0$ be a sharing parameter and β be a learning ratio,

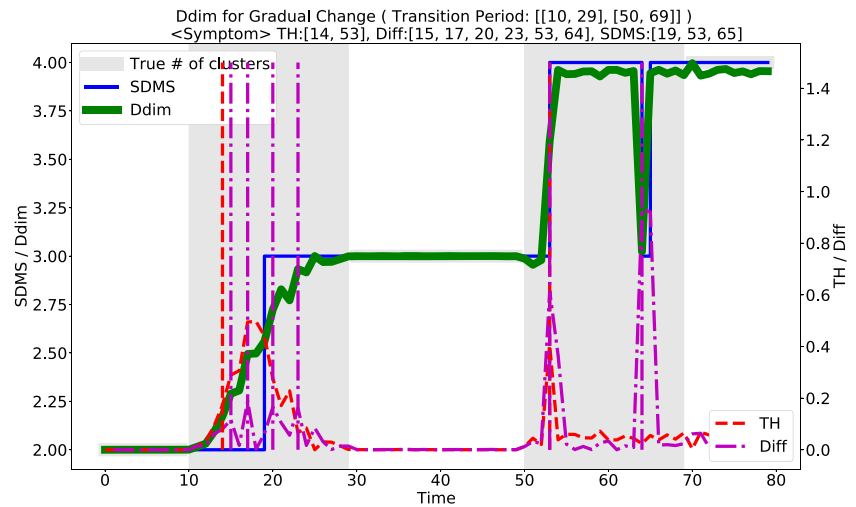
$$w_{t-1,k}^u = w_{t-1,k}^s \cdot \exp\{-\beta L_k(\mathbf{z}_{t-1})\},$$

$$w_{t,k}^s = (1 - \alpha)w_{t-1,k}^u + \sum_{\ell \neq k} \frac{\alpha}{n-1} w_{t-1,\ell}^u,$$

where n is the total number of experts.

Let \hat{k} be the best expert in which $w_{t,k}^s$ is maximum. FS raises an alarm when the best expert changes. The learning rate was set to be the same as our method.

Fig. 5 Ddim graph (transition periods: $[\tau_1 = 9, \tau_2 = 29], [\tau_3 = 49, \tau_4 = 59], T = 79$) Ddim successfully visualizes how rapidly the GMM structure changes in the transition periods from $t = 10$ to 29 and from $t = 50$ to $t = 69$. We see also that Ddim detected signs of model change earlier than SDMS made an alarm of model change



3) *Fixed share weighted algorithm* (FSW-TH, FSW-Diff): We consider variants of TH and Diff where $p(k | \mathbf{y}_t)$ as in (20) is replaced with the normalized weight for k calculated in the process of FS. FSW-TH and FSW-Diff calculate scores by plugging $w_{t,k}^s$ to $p(k | \mathbf{y}_t)$ in (24) and make alarms according to (26) and (27), respectively. The learning rate was set to be the same as our method.

4) *Structural entropy* (SE): It is a measure of uncertainty for model selection, developed in [16]. It is calculated as the entropy with respect to the model posterior probability distribution (20). SE makes alarms when it exceeds a threshold.

The method 1) is the only existing work that performs on-line dynamic model selection. The methods 2) and 3) are the ones that are adapted to our problem setting. The methods 1) and 2) are model change detection algorithms while the method 3) is an algorithm for quantifying latent gradual changes in a similar way with TH or Diff. The method 4) is to detect change signs from the view of model uncertainty, but not to intend continuous model selection.

4.1.4 Results

We generated random data 10 times and took an average value of benefit over 10 trials for each method. Table 3 shows results on comparison of all the methods in terms of AUC both for single and multiple change cases. AUC was calculated for the benefit-FAR curve obtained by varying a threshold. The parameter α specifies the speed of change. As for the multiple change cases, AUC was calculated as an average taken over all change points. Both for the single and multiple change cases, TH and Diff had much higher benefit than the FS-based methods for all the cases. It was statistically significant via t-test with p-values less than 5%. This implies that TH and Diff were able to detect signs of model changes significantly earlier than the FS-based methods. TH worked almost as well as Diff.

It is worthwhile noting that TH and Diff performed better than FSW-TH and FSW-Diff. It implies that the posterior based on the NML distribution is more suitable for tracking gradual model changes than that based on the FS-based heuristics. As α becomes small, the superiority of TH and Diff over the others becomes more remarkable. This implies that Ddim is able to catch up the growth of a cluster much more quickly than the others.

Both for the single and multiple change cases, TH worked slightly better than Diff, but they were almost comparable.

TH and Diff were comparable to SE in terms of change sign detection alone. However, note that the Ddim based ones realize both model change sign detection and continuous

Table 3 AUC comparison results for GMMs. The bold values show highest AUC records over all the methods

Method	$\alpha = 0.2$	$\alpha = 0.5$	$\alpha = 1.0$	$\alpha = 2.0$
(a) Single change point				
TH	0.995	0.920	0.845	0.802
Diff	0.995	0.907	0.830	0.797
SDMS	0.850	0.797	0.775	0.757
FS	0.755	0.715	0.710	0.710
FSW-TH	0.893	0.813	0.778	0.758
FSW-Diff	0.825	0.778	0.750	0.758
SE	0.995	0.927	0.853	0.801
(b) Multiple change point				
TH	0.998	0.925	0.893	0.870
Diff	0.998	0.920	0.883	0.850
SDMS	0.905	0.831	0.804	0.773
FS	0.856	0.794	0.761	0.739
FSW-TH	0.945	0.874	0.816	0.780
FSW-Diff	0.899	0.855	0.834	0.835
SE	0.997	0.928	0.903	0.867

model selection simultaneously. The latter function is specifically important for understanding how fast model changes. Meanwhile, SE can do only sign detection by measuring the uncertainty in model selection. Therefore, Ddim has an advantage over SE in the sense that it can continuously quantify the model complexity in the transition period as well as detects signs of model changes.

4.2 Synthetic data: Auto-regression model

4.2.1 Data sets

We next examined continuous model selection for auto-regression (AR) models as in Section 3.3. We let $n = 1000$ at each time. We generated DataSet 3 according to AR model in the setting as in Section 3.3 where the number k of coefficients in AR model changed over time as follows:

$$k = \begin{cases} 1 & \text{if } 1 \leq t < \tau_1, \\ 1 \text{ with prob. } 1 - \frac{t-\tau_1}{\tau_2-\tau_1} & \text{if } \tau_1 \leq t < \tau_2, \\ 3 \text{ with prob. } \frac{t-\tau_1}{\tau_2-\tau_1} & \text{if } \tau_1 \leq t < \tau_2, \\ 3 & \text{if } \tau_2 \leq t \leq T, \end{cases}$$

where $\tau_1 = 100$, $\tau_2 = 200$ and $T = 300$.

4.2.2 Results

We generated random data 10 times and took an average value of benefit over 10 trials for each method. Table 4 shows results on comparison of all the methods in terms of AUC for Benefit-FAR curves.

Table 4 shows that TH and Diff were comparable to SE and obtained much larger values of AUC than other methods except SE. This implies that TH and Diff could catch up signs of model changes significantly earlier than the others except SE. Note again that TH and Diff conduct not only change sign detection but also continuous model selection meanwhile SE only perform change sign detection.

Table 4 AUC comparison results for AR models. The bold value shows the highest AUC over all the methods

Method	AUC
TH	0.897
Diff	0.896
SDMS	0.771
FS	0.500
FSW-TH	0.771
FSW-Diff	0.500
SE	0.896

4.3 Real data: Market data

4.3.1 Data sets

We apply our method to real market data provided by HAKUHODO,INC. (<https://www.hakuhodo-global.com/>) and M-CUBE,INC. (<https://www.m-cube.com/>). This data set consists of 912 customers' beer purchase transactions from Nov. 1st 2010 to Jan. 31st 2011. See [38]. Each customer's record is specified by a four-dimensional feature vector, each component of which shows a consumption volume for a certain beer category. Categories are: {Beer(A), Low-malt beer(B), Other brewed-alcohol(C), Liquor (D)}.

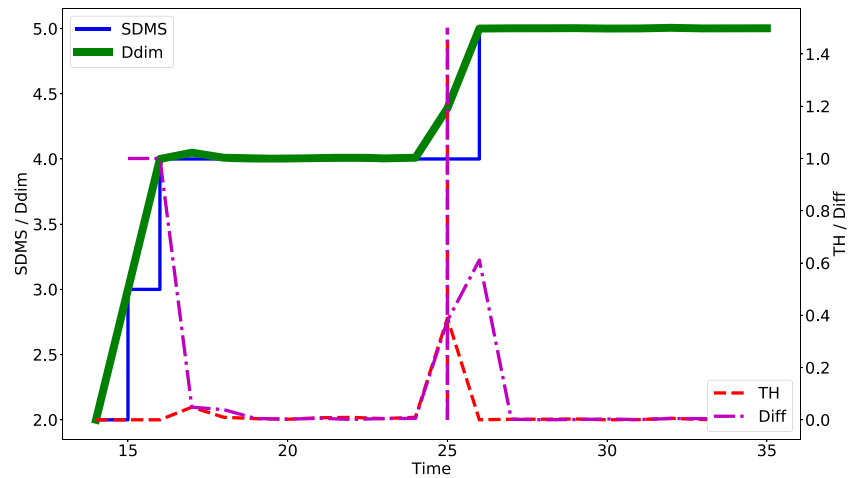
We constructed a sequence of customers' feature vectors as follows: A time unit is a day. At each time $t (= \tau, \dots, T)$, we denote the feature vector of the i th customer as $x_{it} = (x_{it,A}, \dots, x_{it,D}) \in \mathbb{R}^4$. Each $x_{it,j}$ is the i th customer's consumption of the j th category from time $t - \tau + 1$ to t . We denote data at time t as $\mathbf{x}_t = (x_{1t}, \dots, x_{nt})$, where $n = 912$, the number of customers. The total number of transactions is 13993. We set $\tau = 14$ and $T = 53$. Since TH and DIFF have turned out to outperform the other methods in the previous section and we like to conduct continuous model selection simultaneously, we focus on evaluating how well they work for the real data sets.

4.3.2 Results

Figure 6 shows Ddim (green), estimated number of clusters in GMM (blue) using SDMS, and time points of alarms raised by TH and Diff (red and purple) with $\delta_1 = \delta_2 = 0.1$. Table 5 shows the clustering structures $t = 24, 25, 26$. Each number in the (i, j) th cell shows the purchase volume of category $i (= A, B, C, D)$ for the customers in the j th cluster $c_j (j = 1, 2, 3, 4)$. The last row shows the number of customers.

The purchase volume of category C in cluster c_4 gradually increased from $t = 24$ to $t = 25$, eventually c_4 started to collapse at $t = 25$ and was split into c_4 and c_5 at $t = 26$. We confirm from Table 5 that c_4 consisted of heavy users in category C , at $t = 26$, some of them became dormant users that did not purchase anything to form a new cluster. The SDMS algorithm detected this market structure change at $t = 26$. As shown in Fig. 6, TH and Diff successfully raised an alarm at $t = 25$ as a sign of that market structure change. The reason why we could detect the early warning signal is that there were gradual changes among clusters as well as within individual clusters before the clustering change occurred. Our result shows that our method was effective in detecting signs of model changes for such a case.

Fig. 6 Change sign detection for market data. Ddim continuously increases from $t = 24$ to $t = 26$. TH and Diff raised an alarm at $t = 25$ as a sign of that market structure change



4.4 Real Data: Electric power consumption data

4.4.1 Data sets

Next we apply our method to the household electric power consumption dataset provided by [6]. This dataset contains three categories of electric power consumption correspond-

ing to electricity consumed 1) in kitchen and laundry rooms, 2) by electric water heaters and 3) by air-conditioners. The data were obtained every other minute from Dec. 17, 2006 to Dec. 10, 2010. We set $\mathbf{x}_t = (x_1, \dots, x_n)$ and $x_i = (x_{i1}, x_{i2}, x_{i3})$ where each x_i denotes the value of consumption per an hour for the three categories, respectively, and \mathbf{x}_t is the value of consumption for two weeks ($n = 336$).

Table 5 Market structure change

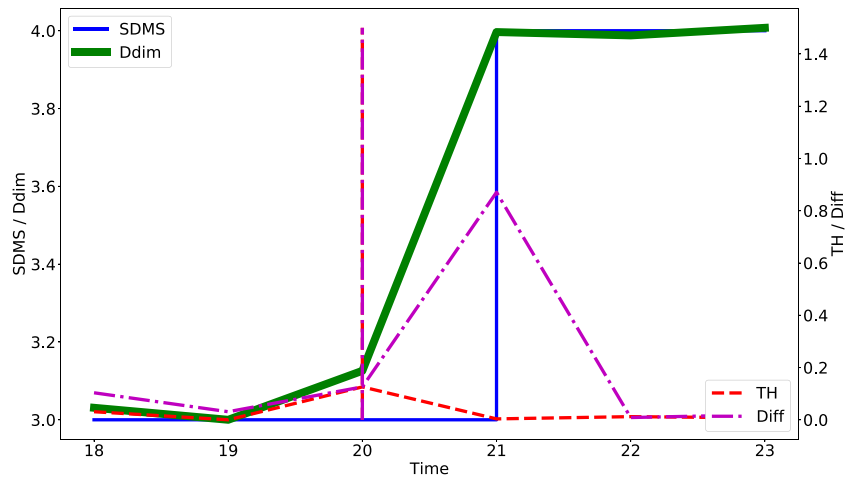
$t = 24$					
cat.	c1	c2	c3	c4	
A	0	1	1993	0	
B	0	2146	0	0	
C	12	25	7	156	
D	1768	1	7	0	
#	211	126	138	437	
$t = 25$					
cat.	c1	c2	c3	c4	
A	0	1	1959	0	
B	0	2201	0	0	
C	11	21	8	184	
D	1919	1	9	0	
#	212	124	141	435	
$t = 26$					
cat.	c1	c2	c3	c4	c5
A	0	0	2139	0	0
B	0	2199	1	0	0
C	12	19	8	2962	0
D	1916	1	12	0	0
#	213	123	148	283	145

4.4.2 Results

Figure 7 shows how Ddim (the green curve) and the number of clusters (the blue line) changed over time. Here each cluster shows a consumption pattern. The red dotted line shows the alarm positions for TH and Diff with $\delta_1 = \delta_2 = 0.1$. Let us focus on the duration from $t = 18$ to $t = 22$. At $t = 18, 19$, there were three clusters, one of which collapsed to two clusters at $t = 21$, eventually, produced the fourth cluster. The Ddim graph in Fig.7 shows that Ddim gradually increased from $k = 3$ to $k = 4$ during the period. The alarm was made by TH and Diff at $t = 20$ while there were still three clusters. This alarm can be thought of as a sign of the emergence of a new cluster having a unique consumption pattern.

Table 6 shows the contents of clusters on the weeks starting from May 14th, 21st, and 28th in 2007. c means clusters, and m_1, m_2, m_3 mean the mean amounts of meter 1,2,3, respectively. The last column shows the total amount of users in a respective cluster. A sign of model change was detected on May 21st. The model change was detected on May 28th. We see from Table 6 that cluster 2 collapsed into clusters 2 and 3. Cluster 2 shows a pattern of homogeneous consumption with a relatively high weight on category 3. Cluster 3 shows a pattern of homogeneous consumption with a relatively high weight on category 1. The sign of this collapse was successfully detected on May 21st by monitoring the Ddim value. The reason why we could detect the early warning signal is that there was a gradual change in the collapse of cluster 2

Fig. 7 Change sign detection for power consumption data. Ddim continuously grows from $t = 19$ and $t = 21$. TH and Diff made an alarm at $t = 20$, which can be thought of as a sign of the emergence of a new cluster with a unique consumption pattern



before the clustering change occurred. Our result shows that our method was effective in detecting signs of model changes for such a case.

5 Relation of Ddim to MDL Learning

This section gives a theoretical foundation of Ddim by relating it to the rate of convergence of the MDL learning algorithm [2, 30]. It selects a model with the shortest total codelength required for encoding the data as well as the model itself. We give an NML-based version of the MDL algorithm as follows.

Let $\mathcal{F} = \{\mathcal{P}_1, \dots, \mathcal{P}_s\}$ where $|\mathcal{F}| = s < \infty$ and each \mathcal{P}_i is a class of probability distributions. For a given training

data sequence $\mathbf{x} = x_1, \dots, x_n$ where each x_i is independently drawn, the MDL learning algorithm selects $\hat{\mathcal{P}}$ such that

$$\begin{aligned} \hat{\mathcal{P}} &= \operatorname{argmin}_{\mathcal{P} \in \mathcal{F}} (-\log p_{\text{NML}}(\mathbf{x}; \mathcal{P})) \\ &= \operatorname{argmin}_{\mathcal{P} \in \mathcal{F}} \left\{ -\log \max_{p \in \mathcal{P}} p(\mathbf{x}) + \log C_n(\mathcal{P}) \right\}, \end{aligned} \tag{31}$$

where $C_n(\mathcal{P})$ is the parametric complexity of \mathcal{P} as in (5). The MDL learning algorithm outputs the NML distribution associated with $\hat{\mathcal{P}}$ as in (31): For a sequence $\mathbf{y} = y_1, \dots, y_n$,

$$\hat{p}(\mathbf{y}) = \frac{\max_{p \in \hat{\mathcal{P}}} p(\mathbf{y})}{C_n(\hat{\mathcal{P}})}. \tag{32}$$

Note that \mathbf{y} is independent of the training sequence \mathbf{x} used to obtain $\hat{\mathcal{P}}$. In previous work [2, 30], the MDL learning algorithm has been designed so that it outputs the two-stage shortest codelength distribution with quantized parameter values, belonging to the model classes. Our algorithm differs from them in that it outputs the NML distribution (32), which is not included in the model classes. The NML distribution and the MDL principle are the central notions in deriving Ddim throughout this paper. Thus it is significant to investigate the relation of Ddim to the NML distribution estimated with the MDL learning algorithm.

We have the following theorem relating Ddim to the rate of convergence of the MDL learning algorithm.

Theorem 3 *Suppose that each \mathbf{x} is generated according to $p^* \in \mathcal{P}^* \in \mathcal{F} = \{\mathcal{P}_1, \dots, \mathcal{P}_s\}$. Let \hat{p} be the output of the MDL learning algorithm as in (32). Let $d_B^{(n)}(\hat{p}, p^*)$ be the Bhattacharyya distance between \hat{p} and p^* :*

$$d_B^{(n)}(\hat{p}, p^*) \stackrel{\text{def}}{=} -\frac{1}{n} \log \sum_{\mathbf{y}} (p^*(\mathbf{y}) \hat{p}(\mathbf{y}))^{\frac{1}{2}}. \tag{33}$$

Table 6 Electric power consumption structure change

The week from May 14th				
c	m1	m2	m3	#
0	0.01	2.73	0.01	163
1	0.00	2.80	6.16	92
2	5.18	3.05	4.74	57
The week from May 21st				
c	m1	m2	m3	#
0	0.01	2.58	0.01	168
1	0.00	2.49	6.22	99
2	5.65	2.64	4.34	45
The week from May 28th				
c	m1	m2	m3	#
0	0.01	2.70	0.01	150
1	0.00	3.01	6.38	99
2	1.73	2.72	5.87	8
3	6.20	3.11	4.91	55

Then for any $\epsilon > 0$, we have the following upper bound on the probability that under the condition for \mathcal{P}^* as in Theorem 1, the Bhattacharyya distance between the output of the MDL learning algorithm and the true distribution exceeds ϵ :

$$\text{Prob}[d_B^{(n)}(\hat{p}, p^*) > \epsilon] = O\left(n^{\text{Ddim}(\mathcal{P}^*)/4} e^{-n\epsilon}\right). \tag{34}$$

Suppose that \mathcal{P} is chosen randomly according to the probability distribution $\pi(\mathcal{P})$ over $\mathcal{F} = \{\mathcal{P}_1, \dots, \mathcal{P}_s\}$ and that the unknown true distribution p^* is chosen from \mathcal{P}^* . Then we have the following upper bound on the expected Bhattacharyya distance between the output of the MDL learning algorithm and the true distribution:

$$E_{\mathcal{P}^*} E_{\mathbf{x} \sim p^* \in \mathcal{P}^*} [d_B^{(n)}(\hat{p}, p^*)] = O\left(\frac{\text{Ddim}(\mathcal{F}^\odot) \log n}{n}\right), \tag{35}$$

where $\text{Ddim}(\mathcal{F}^\odot)$ is Ddim for model fusion as in (16).

The proof is given in Appendix. This result may be generalized into the agnostic case where the model class misspecifies the true distribution (see also [4] for this case). We omit this result from this manuscript since our main concern is how the expected generalization performance is related to Ddim .

Theorem 3 implies that the NML distribution with model of the shortest NML codelength converges exponentially to the true distribution in probability as n increases and the rate is governed by Ddim for the true model. In conventional studies on PAC (probably approximately correct) learning [12], the performance of the empirical risk minimization algorithm has been analyzed using the technique of *uniform convergence*, where the rate of convergence is governed by the metric dimension. Meanwhile, the performance of the MDL learning algorithm is analyzed using the *non-uniform convergence* technique, since the non-uniform model complexity is considered. In this case the rate of convergence of the MDL algorithm is governed by Ddim . Then the expected Bhattacharyya distance between the true distribution and the output of the MDL learning algorithm is characterized by Ddim for model fusion over \mathcal{F} .

6 Conclusion

This paper has proposed a novel methodology for detecting signs of model changes from a data stream. The key idea is to conduct continuous model selection using the notion of descriptive dimensionality (Ddim). Ddim quantifies the real-valued model dimensionality in the model transition period. We are able not only to visualize the model complexity in the transition period of model changes, but also to detect

their signs by tracking the rise-up/descent of Ddim . Focusing on the model changes in Gaussian mixture models, we have shown that gradual structure changes of GMMs can be effectively visualized by drawing a Ddim graph. Furthermore, we have empirically demonstrated that our methodology was able to detect signs of changes of the number of mixtures in GMM and those of the order of AR model earlier than they were actualized. Experimental results have shown that it was able to detect them significantly earlier than any other existing dynamic model selection methods.

This paper has offered the use of continuous model change selection in the scenario of model change sign detection only. Exploring other scenarios of continuous model selection has remained for future studies.

A Proof of Theorem 1

Let \mathcal{P} be a k -dimensional parametric class, which we denote as $\mathcal{P}_k = \{p(\mathbf{x}; \theta) : \theta \in \Theta_k\}$ where Θ_k is a k -dimensional parametric space. In this case, we denote $g(\hat{\theta}, \theta)$ instead of $g(\hat{p}, p)$. Let the finite set of k -dimensional real-valued parameters space be $\bar{\Theta}_k = \{\theta_1, \theta_2, \dots\}$ and let $\bar{\mathcal{P}}_k = \{p(\mathbf{x}; \theta) : \theta \in \bar{\Theta}_k \subset \mathbb{R}^k\}$. Let $I_n(\theta)$ be the Fisher information matrix at θ : $I_n(\theta) \stackrel{\text{def}}{=} (1/n) E_\theta[\partial^2(-\log p(\mathbf{x}; \theta))/\partial\theta\partial\theta^\top]$ and suppose that $\lim_{n \rightarrow \infty} I_n(\theta) = I(\theta)$ for each θ . Below we denote $p(\mathbf{x}; \theta)$ as p_θ . Consider ϵ^2 -neighborhood of θ_i with respect to the KL divergence d_n :

$$D_\epsilon(i) \stackrel{\text{def}}{=} \{\theta : d_n(p_{\theta_i}, p_\theta) \leq \epsilon^2\}.$$

Note that $d_n(p_{\theta_i}, p_\theta)$ is written using Taylor’s expansion up to the second order as follows: Under the condition that $\log p$ is three-times differentiable, $\max_{a,b,c} |\partial^3 \log p(\mathbf{x}; \theta)/\partial\theta_a \partial\theta_b \partial\theta_c| < \infty$,

$$\begin{aligned} d_n(p_{\theta_i}, p_\theta) &= -\frac{1}{n} E_{\theta_i} \left[\frac{\partial \log p(\mathbf{x}; \theta)}{\partial \theta} \Big|_{\theta_i} \right] (\theta - \theta_i) \\ &\quad + \frac{1}{2n} (\theta - \theta_i)^\top E_{\theta_i} \left[-\frac{\partial^2 \log p(\mathbf{x}; \theta)}{\partial \theta \partial \theta^\top} \Big|_{\theta_i} \right] (\theta - \theta_i) \\ &\quad + O(\|\theta - \theta_i\|^3) \\ &= \frac{1}{2} (\theta - \theta_i)^\top I_n(\theta_i) (\theta - \theta_i) + O(\|\theta - \theta_i\|^3), \end{aligned}$$

where we have used the fact:

$$\begin{aligned} E_{\theta_i} \left[\frac{\partial \log p(\mathbf{x}; \theta)}{\partial \theta} \Big|_{\theta_i} \right] &= \sum_{\mathbf{x}} p(\mathbf{x}; \theta_i) \frac{\partial \log p(\mathbf{x}; \theta)}{\partial \theta} \Big|_{\theta_i} \\ &= \frac{\partial \sum_{\mathbf{x}} p(\mathbf{x}; \theta)}{\partial \theta} \Big|_{\theta_i} \\ &= 0. \end{aligned}$$

Therefore, we may consider $\tilde{D}_\epsilon(i)$ in place of $D_\epsilon(i)$.

$$\tilde{D}_\epsilon(i) = \{\theta : (\theta - \theta_i)^\top I_n(\theta_i)(\theta - \theta_i) \leq C\epsilon^2\},$$

where C does not depend on n nor ϵ . Let $B_\epsilon(i)$ be the largest hyper-rectangle within $\tilde{D}_\epsilon(i)$ centered at θ_i . For some $1 \leq C' < \infty$, for any i , we have

$$|B_\epsilon(i)| \leq |\tilde{D}_\epsilon(i)| \leq C' |B_\epsilon(i)|. \tag{36}$$

Along with [25] (p.74), geometric analysis of $B_\epsilon(i)$ yields the Lebesgue volume of $B_\epsilon(i)$ as follows:

$$|B_\epsilon(i)| = \left(\frac{4C\epsilon^2}{k}\right)^{\frac{k}{2}} |I_n(\theta_i)|^{-\frac{1}{2}} = 2^k \prod_{j=1}^k \sqrt{\frac{C\epsilon^2}{k\lambda_j}},$$

where λ_j is the j -th largest eigenvalue of $I_n(\theta_i)$.

We choose $\bar{\Theta}_k$ so that the central limit theorem holds. Then for sufficiently large n , as $\theta \rightarrow \theta_i$,

$$g(\theta_i, \theta) \simeq \left(\frac{n}{2\pi}\right)^{\frac{k}{2}} |I(\theta_i)|^{\frac{1}{2}} e^{-n(\theta - \theta_i)^\top I_n(\theta_i)(\theta - \theta_i)/2}.$$

Thus for $\hat{\theta} \in D_i(\epsilon)$, for sufficiently large n , as $\theta \rightarrow \theta_i$, we obtain

$$g(\hat{\theta}, \hat{\theta}) |B_\epsilon(i)| \simeq \left(\frac{2C\epsilon^2 n}{k\pi}\right)^{\frac{k}{2}}. \tag{37}$$

Next define $Q_\epsilon(i)$ as

$$Q_\epsilon(i) \stackrel{\text{def}}{=} \int_{\hat{\theta} \in \tilde{D}_\epsilon(i)} g(\hat{\theta}, \hat{\theta}) d\hat{\theta}.$$

and let $m_n(\epsilon : \mathcal{P})$ be the smallest number of elements in $\bar{\Theta}_k$:

$$\log C_n(k) = \log \sum_{i=1}^{m_n(\epsilon : \mathcal{P})} Q_\epsilon(i) + O(1). \tag{38}$$

Combining (37) and (36) with (38) yields

$$\log C_n(\mathcal{P}_k) = \log m_n(\epsilon : \mathcal{P}) + \sup_{\bar{\Theta}_k} \left\{ \frac{k}{2} \log \left(\frac{2C\epsilon^2 n}{k\pi} \right) \right\} + O(1) \tag{39}$$

$$= \log m_n(\epsilon : \mathcal{P}) + \frac{k}{2} \log(\epsilon^2 n) + O(1). \tag{40}$$

where C in (39) depends on $\bar{\Theta}_k$, and the $O(1)$ term in (40) may depend on k , but both of them do not depend on n nor

ϵ . The supremum in (39) is taken with respect to $\bar{\Theta}_k$ so that (37) holds. Setting $\epsilon^2 n = O(1)$ yields

$$\begin{aligned} \log C_n(\mathcal{P}_k) &= \log m_n(\epsilon : \mathcal{P}_k) + \frac{k}{2} \log(\epsilon^2 n) + O(1) \\ &= \log m_n(1/\sqrt{n} : \mathcal{P}_k) + O(1). \end{aligned}$$

This completes the proof of Theorem 1. □

B Proof of Theorem 3

Let p^* be the true distribution associated with the true model \mathcal{P}^* . Let $\hat{\mathcal{P}}$ be the model selected by the MDL learning algorithm and let $p_{\text{NML}}(\mathbf{x}; \hat{\mathcal{P}})$ be the NML distribution associated with $\hat{\mathcal{P}}$. We write it as \hat{p} . We employ the proof technique similar to that for two-part code estimators in [2, 30].

By the definition of the MDL learning algorithm, we have

$$\begin{aligned} \min_{\mathcal{P}} (-\log p_{\text{NML}}(\mathbf{x}; \mathcal{P})) &\leq -\log p_{\text{NML}}(\mathbf{x}; \mathcal{P}^*) \\ &= -\log \max_{p \in \mathcal{P}^*} p(\mathbf{x}) + \log C_n(\mathcal{P}^*) \\ &\leq -\log p^*(\mathbf{x}) + \log C_n(\mathcal{P}^*). \tag{41} \end{aligned}$$

Let $p_{\text{NML}, \mathcal{P}}$ be the NML distribution $p_{\text{NML}}(\mathbf{x}; \mathcal{P})$ associated with \mathcal{P} . For $\epsilon > 0$, the following inequalities hold:

$$\begin{aligned} & \text{Prob} \left[d_B^{(n)}(\hat{p}, p^*) > \epsilon \right] \\ & \leq \text{Prob} \left[\mathbf{x} : (41) \text{ holds under } d_B^{(n)}(\hat{p}, p^*) > \epsilon \right] \\ & = \text{Prob} \left[\mathbf{x} : \min_{\mathcal{P} : d_B^{(n)}(p_{\text{NML}, \mathcal{P}}, p^*) > \epsilon} (-\log p_{\text{NML}}(\mathbf{x}; \mathcal{P})) \right. \\ & \quad \left. \leq -\log p^*(\mathbf{x}) + \log C_n(\mathcal{P}^*) \right] \\ & = \text{Prob} \left[\mathbf{x} : \max_{\mathcal{P} : d_B^{(n)}(p_{\text{NML}, \mathcal{P}}, p^*) > \epsilon} p_{\text{NML}}(\mathbf{x}; \mathcal{P}) \geq \frac{p^*(\mathbf{x})}{C_n(\mathcal{P}^*)} \right] \\ & \leq \sum_{\mathcal{P} \in \mathcal{F}, d_B^{(n)}(p_{\text{NML}, \mathcal{P}}, p^*) > \epsilon} \text{Prob} \left[\mathbf{x} : p_{\text{NML}}(\mathbf{x}; \mathcal{P}) \geq \frac{p^*(\mathbf{x})}{C_n(\mathcal{P}^*)} \right]. \tag{42} \end{aligned}$$

Let $E_n(\mathcal{P})$ be the event: $p_{\text{NML}}(\mathbf{x}; \mathcal{P}) \geq p^*(\mathbf{x})/C_n(\mathcal{P}^*)$. Under $E_n(\mathcal{P})$,

$$1 \leq \left(\frac{p_{\text{NML}}(\mathbf{x}; \mathcal{P})}{p^*(\mathbf{x})} \right)^{\frac{1}{2}} (C_n(\mathcal{P}^*))^{\frac{1}{2}}. \tag{43}$$

Then under the condition that $d_B^{(n)}(p_{\text{NML}, \mathcal{P}}, p^*) > \epsilon$, we have

$$\begin{aligned} \text{Prob}[E_n(\mathcal{P})] &= \sum_{\mathbf{x} \cdots E_n(\mathcal{P})} p^*(\mathbf{x}) \\ &\leq \sum_{\mathbf{x} \cdots E_n(\mathcal{P})} p^*(\mathbf{x}) \left(\frac{p_{\text{NML}}(\mathbf{x}; \mathcal{P})}{p^*(\mathbf{x})} \right)^{\frac{1}{2}} (C_n(\mathcal{P}^*))^{\frac{1}{2}} \\ &\leq \left\{ \sum_{\mathbf{y}} (p_{\text{NML}}(\mathbf{y}; \mathcal{P}) p^*(\mathbf{y}))^{\frac{1}{2}} \right\} (C_n(\mathcal{P}^*))^{\frac{1}{2}} \\ &< \exp(-n\epsilon + (\log C_n(\mathcal{P}^*))/2), \end{aligned} \tag{44}$$

where we have used the fact that under $d_B^{(n)}(p_{\text{NML}, \mathcal{P}}, p^*) > \epsilon$, it holds

$$\sum_{\mathbf{y}} (p_{\text{NML}}(\mathbf{y}; \mathcal{P}) p^*(\mathbf{y}))^{\frac{1}{2}} < e^{-n\epsilon}.$$

Plugging (44) into (42) yields

$$\begin{aligned} \text{Prob}[d_B^{(n)}(\hat{p}, p^*) > \epsilon] &\leq \sum_{\mathcal{P} \in \mathcal{F}, d_B^{(n)}(p_{\text{NML}, \mathcal{P}}, p^*) > \epsilon} \text{Prob}[E_n(\mathcal{P})] \\ &< \sum_{\mathcal{P} \in \mathcal{F}, d_B^{(n)}(p_{\text{NML}, \mathcal{P}}, p^*) > \epsilon} \exp(-n\epsilon + (1/2) \log C_n(\mathcal{P}^*)) \\ &\leq \sum_{\mathcal{P} \in \mathcal{F}} \exp(-n\epsilon + (1/2) \log C_n(\mathcal{P}^*)) \\ &= \exp(-n\epsilon + (1/2) \log C_n(\mathcal{P}^*) + \log |\mathcal{F}|). \end{aligned} \tag{45}$$

Under the condition for \mathcal{P}^* as in Theorem 3, we have

$$\frac{1}{2} \log C_n(\mathcal{P}^*) + \log |\mathcal{F}| = \frac{1}{4} \text{Ddim}(\mathcal{P}^*) \log n + o(\log n). \tag{46}$$

Plugging (46) into (45) yields (34).

Let $r_n(\mathcal{P}^*) \stackrel{\text{def}}{=} \{(1/2) \log C_n(\mathcal{P}^*) + \log |\mathcal{F}|\}/n$. For fixed \mathcal{P}^* , we have the following upper bound on the expected Bhattacharyya distance:

$$\begin{aligned} E_{\mathbf{x} \sim p^* \in \mathcal{P}^*} [d_B^{(n)}(\hat{p}, p^*) - r_n(\mathcal{P}^*)] &= \int_0^\infty \text{Prob}[d_B^{(n)}(\hat{p}, p^*) - r_n(\mathcal{P}^*) > \epsilon] d\epsilon \\ &\leq \int_0^\infty e^{-n\epsilon} d\epsilon = \frac{1}{n}, \end{aligned} \tag{47}$$

where we have used (34) to derive (47). Therefore, we have

$$E_{\mathbf{x} \sim p^* \in \mathcal{P}^*} [d_B^{(n)}(\hat{p}, p^*)] \leq r_n(\mathcal{P}^*) + \frac{1}{n}.$$

Taking the expectation with respect to \mathcal{P}^* yields

$$\begin{aligned} E_{\mathcal{P}^*} [d_B^{(n)}(\hat{p}, p^*)] &\leq E_{\mathcal{P}^*} E_{\mathbf{x} \sim p^* \in \mathcal{P}^*} [r_n(\mathcal{P}^*)] + \frac{1}{n} \\ &= O\left(\frac{E_{\mathcal{P}^*} [\log m_n(1/\sqrt{n} : \mathcal{P}^*)]}{n}\right) \\ &= O\left(\frac{E_{\mathcal{P}^*} [\text{Ddim}(\mathcal{P}^*) \log n]}{n}\right) \\ &= O\left(\frac{\text{Ddim}(\mathcal{F}^\odot) \log n}{n}\right). \end{aligned} \tag{48}$$

To derive (48), we have used the fact $\log C_n(\mathcal{P}^*) = \log m_n(1/\sqrt{n} : \mathcal{P}^*) + O(1)$ (by Theorem 1). This completes the proof. \square

Acknowledgements This work was partially supported by JST KAKENHI 19140000190.

Funding Open access funding provided by The University of Tokyo.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Alquier P, Ridgway J, Chopin N (2016) ‘‘On the properties of variational approximations of Gibbs posteriors.’’ *J Mach Learning Research* 17,1–41
- Barron A, Cover T (1991) Minimum complexity density estimation. *IEEE Trans Information Theory* IT 37:1034–1054
- Davis RA, Lee T, Rodriguez-Yam G (2006) ‘‘Structural break estimation for nonstationary time series models.’’ *Jr. of the Amer. Stat. Assoc.* 101(473):22–239
- Ding J, Tarokh V, Yang Y (2018) Model selection techniques: an overview. In: *IEEE Signal Processing Magazine* Vol. 35, Issue: 6, pp 16–34 Nov 2018
- Ding J, Zhou J, Tarokh V (2019) Asymptotically optimal prediction for time-varying data generating processes. In: *IEEE Transactions on Information Theory* Vol. 65, Issue: 5, pp 3034–3067 May 2019
- Dheeru D, Taniskidou EK (2017) UCI machine learning repository
- Dudley RM (1987) Universal Donsker classes and metric entropy. *Annals of Probability* 15(4):1306–1326
- Erven T, Grunwald P, Rooij S (2012) ‘‘Catching up by switching sooner: a predictive approach to adaptive estimation with an application to the AIC-BIC dilemma.’’ *Jr. Royal Stat Soc Ser B* Vol. 74, No. Issue 3 pp 361–417
- Farmer JD (1982) Information dimension and the probabilistic structure of chaos. *Z. Naturforsch A* 31:1304–1325

10. Gama J, Zlibait I, Bifet A, Pechenizkiy M, Bouchachia A (2014) A survey on concept drift adaptation. *ACM Computing Survey* 46(4):1–37
11. Grunwald PD (2007) *The minimum description length principle*, MIT press
12. Haussler D (1992) Decision theoretic generalizations of the PAC model for neural net and other learning applications. *Information and Computation* 100:78–150
13. Herbster M, Warmuth M (1998) Tracking the best experts. *Machine Learning* 32:151–178
14. Hirai S, Yamanishi K (2012) “Detecting changes of clustering structures using normalized maximum likelihood coding.” In: *Proceedings of ACM International Conference on Knowledge Discovery and Data Mining (KDD’12)* pp 343–351
15. Hirai S, Yamanishi K (2019) Correction to efficient computation of normalized maximum likelihood codes for Gaussian mixture models with its applications to clustering. *IEEE Transactions on Information Theory* 65(10):6827–6828
16. Hirai S, Yamanishi K (2018) “Detecting structural uncertainty with structural entropy.” In: *Proceedings of 2018 IEEE International Conference on BigData (BigData’18)* pp 26–35
17. Huang DTJ, Koh YS, Dobbie G, Pears R (2014) “Detecting volatility shift in data streams.” In: *Proceedings of 2014 IEEE International Conference on Data Mining (ICDM’14)*, pp 863–868
18. Kolmogorov AN, Tihomirov VM (1961) ϵ -entropy and ϵ -capacity of sets in functional spaces. *Amer. Math. Soc. Transl. Ser.* 2(17):277–364
19. MacLachlan G, Peel D (2000) *Finite Mixture Models*. John Wiley and Sons
20. Mandelbrot BB (1982) *The Fractal Geometry of Nature*. Freeman, W.H
21. Ntoutsis I, Spiliopoulou MY (2011) “Theodoridis:FINGERPRINT: Summarizing cluster evolution in dynamic environments”. *Proceedings of Computational Science and Its Applications-ICCSA 2011. Part II* 11:562–577
22. Ohsawa Y (2018) Graph-based entropy for detecting explanatory signs of changes in market. In: *The Review of Social Network Strategies Vol. 12, 2*, pp 183–203
23. Pollard D (1984) *Convergence of Stochastic Processes*. Springer-Verlag, Berlin/NewYork
24. Rissanen J (1978) Modeling by shortest description length. *Automatica* 14(5):465–471
25. Rissanen J (2012) *Optimal Estimation of Parameters*. Cambridge
26. Shtarkov YM (1987) Universal sequential coding of single messages. *Problem Peredachi Informatsii* 23(3):3–17
27. Song M, Wang H (2005) “Highly efficient incremental estimation of Gaussian mixture models for online data stream clustering.” *Intelligent Computing: Theory and Applications III* 5803:174–183
28. Vapnik VN (1982) *Estimation of Dependences Based on Empirical Data*. Springer Verlag, New York
29. Xuan X, Murphy K (2007) “Modeling changing dependency structure in multivariate time series.” *Proceedings of the 24th International Conference on Machine Learning* pp 1055–1062 ACM
30. Yamanishi K (1992) “A learning criterion for stochastic rules.” *Machine Learning Volume 9, Issue 2-3*, pp 165–203 July 1992
31. Yamanishi K (2019) “Descriptive dimensionality with its characterization of MDL-based learning and change detection.” [arXiv:1910.11540](https://arxiv.org/abs/1910.11540)
32. Yamanishi K, Hirai S (2023) “Detecting signs of model change with continuous model selection based on descriptive dimensionality.” [arXiv:2302.12127](https://arxiv.org/abs/2302.12127)
33. Yamanishi K, Fukushima S (2018) Model change detection with the MDL principle. *IEEE Transactions on Information Theory* 64(9):6115–6126
34. Yamanishi K, Maruyama Y (2007) Dynamic model selection with its applications to novelty detection. *IEEE Transactions on Information Theory* 53(6):2180–2189
35. Yamanishi K, Maruyama Y (2005) “Dynamic syslog mining for network failure monitoring.” In: *Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD’05)* pp 499–508
36. Yamanishi K, Miyaguchi K (2016) “Detecting gradual changes from data stream using MDL change statistics.” In: *Proceedings of 2016 IEEE International Conference on BigData (BigData’16)* pp 156–163
37. Yamanishi K, Xu L, Yuki R, Fukushima S, Lin C (2021) Change sign detection with differential MDL change statistics and its applications to COVID-19 pandemic analys. *Scientific Reports* 11:19795
38. <https://github.com/conf-post-01/Ddim.git>

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.