Online Bayesian Kernel Segmentation and a application

Shuang Na
Department of Mathematics & Statistics
University of South Florida
Email: sna@mail.usf.edu

Kandethody M. Ramachandran
Department of Mathematics & Statistics
University of South Florida
4202 E Fowler Ave, CMC 342, Tampa, Florida,USA
Email: ram@usf.edu

Ming Ji
College Of Nursing
University of South Florida
Email:mji@health.usf.edu

***Abstract*—In recent years, data mining have been explored in many areas such as statistics, finance, engineering and biology etc. In order to represent the data more efficiently and effectively, one of mining process is supported by time series segmentation. Segmentation is looking for the change points between two different patterns and developing a suitable model depending on observed data. Based on the issue of limited computing and storage capabilities, it is necessary to consider an adaptive and online segmentation method. In this paper, we proposes an online Bayesian Kernel Segmentation method, which considering multivariate density function as predictive distribution instead of calculating posterior predictive distribution. In the empirical human pattern segmentation result, it shows 92% overall segmentation accuracy.**

I. Introduction

Time series analysis have been applied in many fields such as human activity identification ([1],[2],[3]), voice recognition ([4],[5]) and sign language ([6],[7])etc. At this point, the aim of time series analysis is to extract information from the certain time period ([8],[9]), by considering all data points during the time interval as a whole instead of individual points. In addition, these extracted informations can be represented as some target value and their target values may change over time, i.e. the statistical properties of probability function and fitting model may change with time. The important problem of mining streaming scenario is dimension reduction, classification, clustering, frequency counting and segmentation [8], which support to empirical analysis. As one of data mining methods, segmentation algorithm can represent the observation more effectively and efficiently.

Time series segmentation is to break a time series interval into optimal non-overlapping segments automatically and using these segments we will also develop a suitable model based on these observations of each segments simultaneously, such as a regression model and a probability density function. The boundary of two connected segments represents abrupt change. Further more, the segmentation algorithm can be divided into two groups, offline and online algorithm. [10] and [11] has a review on time series segmentation about bottom-up, top-down, sliding window and SWAB (sliding window and bottom-up) algorithm. The most common offline algorithms are top-down and bottom-up algorithm In order to improve accuracy, many papers extend the two offline methods based on different technical skills. [12] introduces a local iterative replacement and global iterative replacement methods required by dynamic programing. Bayesian method has been applied to discover change points [13] by posterior probability. [14] used the fisher information as the cost function rather than error function. However, due to the properties of continuously incoming data, an adaptive and incremental algorithm is more suitable for dealing with time series. For another category of segmentation, sliding window algorithm has been applied for defining segments as an online method. Nonetheless, sliding window gives us undesirable experimental results [15]. There are other several online segmentation algorithms that have been proposed to improve the performance of online segmenting. These algorithms are built on different main ideas, such as Bayesian method ([16],[17],[18]) and HMM ([19],[20], etc.

In this paper, we propose an online Bayesian Kernel Segmentation(OBKS) method that modified Online Bayesian Change point detection [16], which apply online kernel density function as predictive function instead of Bayesian predictive function. One of the advantages of Bayesian approaches is that it considers all uncertainty as prior distribution. Another advantage is that it does not require the asymptotic assumptions about test statistics that are present in frequentist algorithms, which can be problematic in situations where the parametric models considered are restricted to a finite, possibly small intervals of time [21]. However, it's challenging to choose perfect prior function that can be used for many cases. On the other word, it will cost more time if it's far away from true parameters. Also, there are two "prior functions" in [16], which resulted in more difficult to choose correct prior distribution. The detail will be given in section II. Meanwhile, kernel density function is brought out to get away prior function that used to generate Bayesian predictive probability. There are few articles that discussed about multivariate online kernel density estimation algorithms ([22],[23],[24]).

This study is organized as follows. In the following section, a online Bayesian kernel method is proposed. Section III discusses the application of online Bayesian kernel method by considering simulated observation and empirical data. The conclusion is given in section IV.

II. The Segmentation Method

The proposed algorithm is motivated by [16], in which instead of generating posterior predictive density of a new incoming and unknown data based on all already observed datum, we consider online multivariate kernel density based on all already observed points. First, let's briefly introduce the Online Bayesian method.

A. Online Bayesian method

Homogeneous observations from a same segments are assumed to follow a certain distribution, and those heterogeneous observations from disjointed segments follow different distributions. Therefore, to find the change point between two patterns becomes very important problem. Bayesian online detecting method ([16],[25]) consider the concept of "run length" r_t, which is the observation length of the current posterior distribution at time t and it is linear about time t. For example, if $r_t = 0$ at t=6, x_6 is a change point; if $r_t \neq 0$, we keep running one more time and repeat the process. $x_t^{(r)}$ is defined as the set included all observations correspond to run length r_t. If r_t is zero, $x^{(r)}$ is empty set. For example, t=6, $r_t = 1$, then $x_6^{(r)} = \{x_5, x_6\}$. In order to find the posterior distribution $P(r_t|x_{1:t})$, we need generate a recursive joint distribution $P(r_t, x_{1:t})$,

$$\begin{aligned} P(r_t|x_{1:t}) &= \frac{P(r_t, x_{1:t})}{P(x_{1:t})} \propto P(r_t, x_{1:t}) \\ &\propto \sum_{r_{t-1}} P(r_t, r_{t-1}, x_t, x_{1:t-1}) \\ &\propto \sum_{r_{t-1}} P(r_t, x_t|r_{t-1}, x_{1:t-1}) P(r_{t-1}, x_{1:t-1}) \\ &\propto \sum_{r_{t-1}} P(r_t|r_{t-1}) P(x_t|r_{t-1}, x_{t-1}^{(r)}) P(r_{t-1}, x_{1:t-1}) \end{aligned} \tag{1}$$

Here, $P(r_t|r_{t-1})$ is prior probability , the joint distribution $P(r_t, x_{1:t})$ is called growth probability and $P(x_t|r_{t-1}, x_{t-1}^{(r)})$ is predictive probability. At every time recursion, we pick the r_t that's associated with the largest posterior probability, which r_t is also associated with the largest joint distribution in recent data. Therefore, we need to get prior distribution $P(r_t|r_{t-1})$ and the predictive distribution $P(x_t|r_{t-1}, x_t^{(r)})$ to compute posterior distribution.

The prior distribution has two directions: one direction is that no change point happens at time t and $r_t = r_{t-1} + 1$ with probability $1 - H(r_t) = 1 - 1/\lambda$, which means the new data join the current group and follows same distribution; another one is that a change point occurs at time t, r_t drop to 0 with probability $H(r_t) = 1/\lambda$. Here, $H(r_t)$ is hazard function based on geometric distribution with parameter λ [26]. The prior distribution is:

$$P(r_t|r_{t-1}) = \begin{cases} H(r_{t-1}) & \text{if } r_t = 0 \\ 1 - H(r_{t-1}) & \text{if } r_t = r_{t-1} + 1 \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

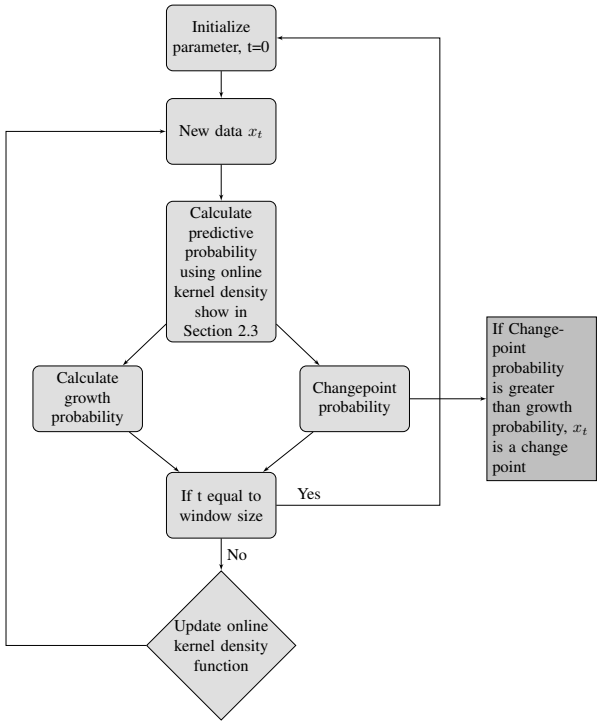

Fig. 1. Online Bayesian Kernel Segmentation

Here, the predictive probability $P(x_{t+1}|r_t, x_t^{(r)})$ is a multivariate kernel density function only depends on the recent data set $x_t^{(r)}$, due to the distribution stays the same if not change point occurs. The more details are discussed in section II(B).

B. Online Multivariate kernel estimation

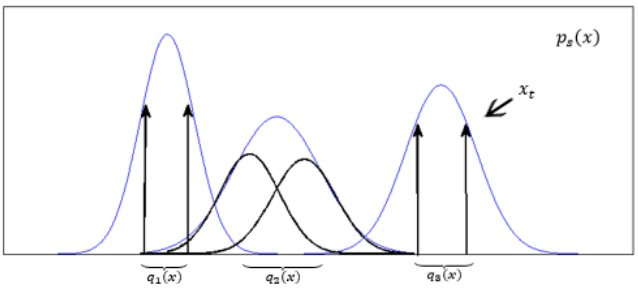

Fig. 2. Sample distribution $p_s(\mathbf{x})$ with associated detail distributions $q_i(x)$

[22] has proposed a Online multivariate kernel (oKDE) density estimation algorithm, which created an online bandwidth estimation method and designed a compression model that reduce the oKDE's complexity. The compressed model of d-dimensional data as an N-component gaussian mixture model is defined as:

$$p_s(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i \phi_{\Sigma_{si}}(\mathbf{x} - \mathbf{x}_i) \tag{3}$$

where

$$\phi_\Sigma(\mathbf{x}-\mu) = (2\pi)^{-d/2}|\Sigma|^{-1/2}e^{-\frac{1}{2}(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)}$$

is Gaussian kernel with center μ and covariance matrix Σ. α_i is weight and $\sum_i \alpha_i = 1$. Kernel density estimation with a bandwidth $\mathbf{H}$(covariance matrix):

$$\hat{p}_{KDE} = \phi_{\mathbf{H}} * p_s(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i \phi_{\Sigma_{si}+\mathbf{H}}(\mathbf{x}-\mathbf{x}_i) \tag{4}$$

In order to reduce the complexity of KDE as new data being added, we need to compress the sample distribution $p_s(\mathbf{x})$ with time by replacing clusters of components. There is an additional model $q_i(\mathbf{x})$ for each component is used for recover from these early over-compressions (Figure 2), therefore the combined model is:

$$S_{model} = \{p_s(\mathbf{x}), \{q_i(\mathbf{x})\}_{i=1:N}\} \tag{5}$$

*1) **Bandwidth selection**:* The classic measure of difference with $\hat{p}_{KDE}$ and unknown underlying pdf is asymptotic mean integrated squared error(AMISE), defined as:

$$AMISE = (4\pi)^{-d/2}|\mathbf{H}|^{-1/2}N_\alpha^{-1} + \frac{1}{4}d^2 \int tr^2\{\mathbf{H}\mathscr{G}_p(\mathbf{x})\}d\mathbf{x} \tag{6}$$

Where $tr(\cdot)$ is the trace operator, $\mathscr{G}_p(\mathbf{x})$ is a Hessian of $p(\mathbf{x})$ and $N_\alpha = (\sum_{i=1}^{N}\alpha_i^2)^{-1}$. If we rewrite the bandwidth matrix in terms of scale β and a known structure $\mathbf{F}$, $\mathbf{H} = \beta^2\mathbf{F}$. Minimize (6) respect to scale is:

$$\beta_{opt} = [d(4\pi)^{d/2}N_\alpha R(p,\mathbf{F})]^{-\frac{1}{d+4}} \tag{7}$$

where

$$R(p,\mathbf{F}) = \int tr^2\{F\mathscr{G}_p(\mathbf{x})\}d\mathbf{x}$$

Usually, this function is estimated by plug-in method [27]. Here, $R(p,\mathbf{F})$ can be written as expectation of the derivatives $\psi_r = \int p^{(r)}(\mathbf{x})p(\mathbf{x})d\mathbf{x}$. We can use the $p_s(\mathbf{x})$ to obtain the approximation:

$$p(\mathbf{x}) \approx p_s(\mathbf{x}), p^{(r)}(\mathbf{x}) = p_{\mathbf{G}}^{(r)}(\mathbf{x}) \tag{8}$$

where we approximate $p_{\mathbf{G}}^{(r)}(\mathbf{x})$, the derivative of $p(\mathbf{x})$ through the kernel density estimation:

$$p_{\mathbf{G}}(\mathbf{x}) = \phi_{\mathbf{G}}(\mathbf{x}) * p_s(\mathbf{x}) = \sum_{j=1}^{N} \alpha_j \phi_{\Sigma_{sj}+\mathbf{G}}(\mathbf{x}-\mu_j) \tag{9}$$

The estimate $p_G(\mathbf{x})$ is called pilot distribution, $\mathbf{G}$ is pilot bandwidth. Combine with approximation in (8), the estimation of $R(p,\mathbf{F})$ is:

$$R(p,\hat{\mathbf{F}},\mathbf{G}) = \int tr\big(\mathbf{F}\mathscr{G}_{p\mathbf{G}}(\mathbf{x})\big)tr\big(\mathbf{F}\mathscr{G}_{ps}(\mathbf{x})\big) \tag{10}$$

To get the functional result (10) using matrix algebra,

$$\begin{aligned} &R(p,\hat{\mathbf{F}},\mathbf{G}) \\ &= \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i\alpha_j\phi_{\mathbf{A}_{ij}^{-1}}(\Delta_{ij}) \\ &\times [2tr(\mathbf{F}^2\mathbf{A}_{ij}^2)(1-2m_{ij}) + tr^2(\mathbf{F}\mathbf{A}_{ij})(1-m_{ij})^2] \end{aligned} \tag{11}$$

where

$$\begin{aligned} \mathbf{A}_{ij} &= (\Sigma_{gi}+\Sigma_{sj})^{-1}, \\ \Delta_{ij} &= \mu_i + \mu_j, \\ m_{ij} &= \Delta_{ij}^T\mathbf{A}_{ij}\Delta_{ij} \end{aligned} \tag{12}$$

We use the empirical covariance of sample observation $\hat{\Sigma}_{smp}$ to estimate F, i.e $\mathbf{F} = \hat{\Sigma}_{smp}$. We estimate pilot bandwidth $\mathbf{G}$ by a multivariate normal-scale rule:

$$\mathbf{G} = \hat{\Sigma}_{smp}\Big(\frac{4}{(d+2)N_\alpha}\Big)^{\frac{2}{d+4}} \tag{13}$$

Where $\hat{\Sigma}_{smp}$ can be updated using recursive covariance matrix rule, $\hat{\Sigma}_t$ and $\hat{\mu}_t$ is covariance and mean for observed t data points:

$$\begin{aligned} \hat{\mu}_t &= \frac{t-1}{t}\hat{\mu}_{t-1} + \frac{1}{t}\mathbf{x}_t \\ \hat{\Sigma}_t &= \frac{t-1}{t}\hat{\Sigma}_{t-1} + \frac{1}{t}(\mathbf{x}_t-\hat{\mu}_{t-1})(\mathbf{x}_t-\hat{\mu}_{t-1})^T \end{aligned} \tag{14}$$

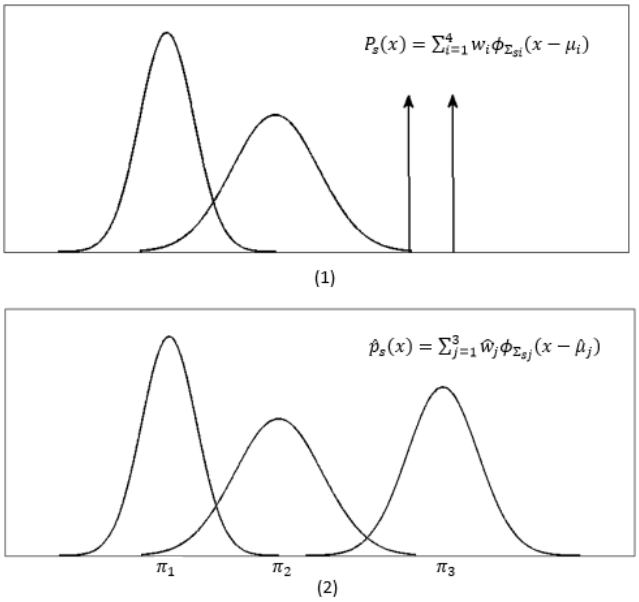

Fig. 3. Compress four components sample distribution $p_s(\mathbf{x})$ (1) into three components sample distribution (2)

*2) **Compression**:* This part introduce compressing (Figure 3)and refining the original N-component sample distribution by a M-component model $\hat{p}_s(\mathbf{x})$, $M < N$. Here, $\hat{w}_j$ is weight

and $\sum_j \hat{w}_j = 1$.

$$\hat{p}_s(\mathbf{x}) = \sum_{j=1}^{M} \hat{w}_j \phi_{\Sigma_{sj}}(\mathbf{x} - \hat{\mu}_j) \tag{15}$$

Because of slow convergence for moderate number of dimensions, there is a clustering-based approach [28], which is to identify clusters of components in $p_s(\mathbf{x})$ and each cluster is associated with a single component.Let $\Xi(M) = \{\pi_j\}_{j=1:M}$ be a collection of disjoint sets of indexes (Figure 2). Therefore,

$$p_s(\mathbf{x}; \pi_j) = \sum_{i \in \pi_j} w_i \phi_{\Sigma_{si}}(\mathbf{x} - \mu_i) \tag{16}$$

The parameters of $j-th$ component are defined as:

$$\begin{aligned} \hat{w}_j &= \sum_{i \in \pi_j} w_i, \hat{\mu}_j = \hat{w}_j^{-1} \sum_{i \in \pi_j} w_i \hat{\mu}_i \\ \hat{\Sigma}_j &= \hat{w}_j \sum_{i \in \pi_j} w_i (\Sigma_i + \mu_i \mu_i^T) - \hat{\mu}_j \hat{\mu}_j^T \end{aligned} \tag{17}$$

Hence, the compression is to identify minimal number of M and the clustering $\Xi(M)$, which construct the lowest clustering error.

$$\hat{M} = argmin_M E(\Xi(M)), s.t. E(\Xi(\hat{M})) \leq D_{th} \tag{18}$$

where D_{th} is pre-defined threshold, $E(\Xi(\hat{M}))$ is largest local clustering error. Here,

$$E(\Xi(\hat{M})) = max_{\pi_j \in \Xi(M)} \hat{E}(p_s(\mathbf{x}; \pi_j), H_{opt}) \tag{19}$$

*3) **Local clustering error**:* We want to approximate component in (16) with a single Gaussian $p_0(\mathbf{x})$using method in (17). The local clustering error is defined as:

$$\hat{E}(p_1(\mathbf{x}), H_{opt}) = D(p_{1KDE}(\mathbf{x}), p_{0KDE}(\mathbf{x})) \tag{20}$$

where,

$$\begin{aligned} &\mathbf{H}_{opt}\text{is current estimated bandwidth} \\ &p_1(\mathbf{x}) = p_s(\mathbf{x}; \pi_j) \\ &p_{1KDE}(\mathbf{x}) = p_1(\mathbf{x}) * \phi_{\mathbf{H}_{opt}}(\mathbf{x}) \\ &p_{0KDE}(\mathbf{x}) = p_0(\mathbf{x}) * \phi_{\mathbf{H}_{opt}}(\mathbf{x}) \end{aligned} \tag{21}$$

In addition, the distance between distribution is using Hellinger distance,

$$\begin{aligned} &D^2(p_{1KDE}(\mathbf{x}), p_{0KDE}(\mathbf{x})) \\ &= \frac{1}{2} \int \left((p_{1KDE}(\mathbf{x})^{1/2} - p_{0KDE}(\mathbf{x}))^{1/2} \right)^2 d\mathbf{x} \end{aligned} \tag{22}$$

Because it cannot be calculated analytically for the mixture model, we use unscented transform ([29],[30])to approximate it.

*4) **compression by hierarchical error minimization**:* A hierarchical approach can be used to optimize (18) with all possible clusters $\Xi(M)$ for the number of clusters M, which start by splitting the entire sample distribution into two sub-mixtures (16) using Goldberger's K-means algorithm. Each sub-mixture will estimate a single Gaussian $p_0(\mathbf{x})$. The hierarchical process is recursively splitting the tree until the largest local error is sufficiently small and satisfy $E(\Xi(M)) \leq D_{th}$.

*5) **online kernel density estimation**:* The first step is to update sample with combine previous model and new observation using weight $w_0 = N_t^{-1}$:

$$\tilde{p}_{s(t)}(\mathbf{x}) = (1 - w_0) p_{s(t-1)}(\mathbf{x}) + w_0 \phi_0(\mathbf{x} - \mathbf{x}_t) \tag{23}$$

Let $\tilde{q}_{i(t)}(\mathbf{x}) = \phi_0(\mathbf{x} - \mathbf{x}_t)$, we have the updated sample model,

$$\tilde{S}_{model(t)} = \{\tilde{p}_{s(t)}(\mathbf{x}), \{\tilde{q}_{i(t)}(\mathbf{x})\}_{i=1:\tilde{M}_t}\} \tag{24}$$

$$\{\tilde{q}_{i(t)}(\mathbf{x})\}_{i=1:\tilde{M}_t} = \{q_i(\mathbf{x})\}_{i=1:M_t} \tag{25}$$

Here, $\tilde{\cdot}$ denotes the update model before the compression. The bandwidth in (13) as updating by $N_{\alpha t} = \left(N_{\alpha(t-1)}^{-1}(1-w_0)^2 + w_0^2\right)^{-1}$. Therefore,

$$\begin{aligned} \mathbf{H}_t &= \mathbf{F}[d(4\pi)^{d/2} N_{\alpha t} \hat{R}(p, \mathbf{F}, \mathbf{G})]^{-\frac{2}{d+4}} \\ \mathbf{F} &= \hat{\Sigma}_{smp} \\ \mathbf{G} &= \hat{\Sigma}_{smp} \Big(\frac{4}{(d+2)N_{\alpha t}} \Big)^{\frac{2}{d+4}} \end{aligned} \tag{26}$$

Summarize above step details, the online Bayesian kernel density estimation (Figure 4)is as follow combined Algorithm 1, Algorithm 2 and Algorithm 3:

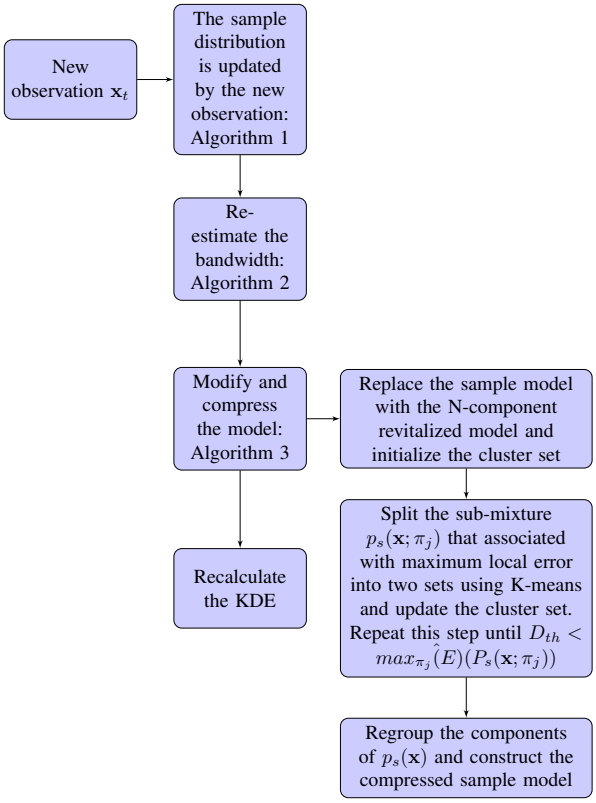

Fig. 4. Online Multivariate Kernel Density Estimation

Algorithm 1 Update the sample model

procedure UPDATE THE SAMPLE MODEL
At time t, the sample is defined as:
$S_{model(t)} = \{p_{st}(\mathbf{x}), \{q_{it}(\mathbf{x})\}_{i=1:M_t}\}$
Update the effective number of observed samples:
$N_{t+1} = N_t + 1$ and $w_0 = 1/N_{t+1}$
Update the sample distribution at time $t+1$:
$\tilde{p}_{s(t+1)}(\mathbf{x}) = (1-w_0)p_{st}(\mathbf{x}) + w_0\phi_0(\mathbf{x} - \mathbf{x}_{t+1})$
The sample model at time $t+1$ become to:
$\tilde{S}_{model(t+1)} = \{\tilde{p}_{s(t+1)}(\mathbf{x}), \{\tilde{q}_{i(t+1)}(\mathbf{x})\}_{i=1:\tilde{M}_t}\}$
where $\{\tilde{q}_{i(t+1)}(\mathbf{x})\}_{i=1:\tilde{M}_t} = \{\{q_{it}(\mathbf{x})\}_{i=1:M_t}, \tilde{q}_{\tilde{M}_t}(\mathbf{x}) = \phi_0(\mathbf{x} - \mathbf{x}_{t+1})\}$
end procedure

Algorithm 2 Update bandwidth

procedure BANDWIDTH ESTIMATION
Update empirical covariance $\hat{\Sigma}_{smp}$ using (16)to approximate the covariance from a single Gaussian
Update $N_{\alpha(t+1)} = (N_{\alpha t}^{-1}(1-w_0)^2 + w_0^2)^{-1}$
Re-calculate $\hat{R}(p, \mathbf{F}, \mathbf{G})$ using (11) and (26)
Estimate the optimal bandwidth at time $t+1$ by (26)
end procedure

Algorithm 3 Compress the sample model

procedure COMPRESS THE SAMPLE MODEL
According to Algorithm 1 and Algorithm 2, $\tilde{S}_{model(t+1)}$ and $\mathbf{H}_t$ is estimated
Re-calculate each i-th component in $\tilde{S}_{model(t+1)}$ when $\hat{E}(\tilde{q}_i(\mathbf{x}), \mathbf{H}_{t+1}) > D_{th}$
Initialize the cluster set: $M = 1, \Xi(M) = \pi_1, \pi_1 = \{1, 2, ..., N\}$
Do until $max_{\pi_j \in \Xi(M)} \hat{E}(p_s(\mathbf{x}; \pi_j)) < D_{th}$
Select the cluster j such that $\pi_j = argmax_{\pi_j \in \Xi(M)} \hat{E}(p_s(\mathbf{x}; \pi_j))$
Split π_j into two sub sets π_{j1} and π_{j2}using the Goldberger's K-means
M=M+1, $\Xi(M) = \{\{\Xi(M)\ \pi_j\}, \pi_{j1}, \pi_{j2}\}$
End loop
Construct each component in $\hat{p}_s(\mathbf{x})$ and its detailed model $\hat{q}_j(\mathbf{x})$ according to the clustering $\Xi(M)$
end procedure

III. EXPERIMENT RESULT

To compare with [16], instead of using Bayesian predictive function as predictive model, we apply online multivariate kernel density estimation as predictive function of the current model when new data coming. Two experiments are manipulated to evaluate the innovated Online Bayesian kernel segmentation method. These experiments compares this proposed method with [16] using simulated data and empirical observation. Here,oBK represents online Bayesian Kernel method and oB represents online Bayesian method.

A. Simulated data experiment

Firstly, to estimate the segmentation accuracy, we generate four different combinations of bi-normal variables using Markov Chain Monte Carlo(MCMC) simulation technique: low covariance with low correlation, low covariance with high correlation, high covariance with low correlation and high covariance with high correlation. Each combination includes three types of bi-normal variables (30). The testing result is showed in TableI.

Combination 1: low covariance with low correlation

$$\mu_1 = (00), \Sigma_1 = \begin{bmatrix} 1.6 & -0.2 \\ -0.2 & 1 \end{bmatrix}$$
$$\mu_2 = (22), \Sigma_2 = \begin{bmatrix} 1 & 0 \\ 0 & 0.5 \end{bmatrix}$$
$$\mu_3 = (-11), \Sigma_2 = \begin{bmatrix} 1 & 0.3 \\ 0.3 & 1 \end{bmatrix} \tag{27}$$

Combination 2: low covariance with high correlation

$$\mu_1 = (00), \Sigma_1 = \begin{bmatrix} 1 & 0.7 \\ 0.7 & 1 \end{bmatrix}$$
$$\mu_2 = (-11), \Sigma_2 = \begin{bmatrix} 1 & 0.7 \\ 0.7 & 1 \end{bmatrix}$$
$$\mu_3 = (2-2), \Sigma_2 = \begin{bmatrix} 1 & 0.9 \\ 0.9 & 1.5 \end{bmatrix} \tag{28}$$

Combination 3: high covariance with low correlation

$$\mu_1 = (00), \Sigma_1 = \begin{bmatrix} 4 & -0.4 \\ -0.4 & 4 \end{bmatrix}$$
$$\mu_2 = (-11), \Sigma_2 = \begin{bmatrix} 3 & 0.3 \\ 0.3 & 5 \end{bmatrix}$$
$$\mu_3 = (2-2), \Sigma_2 = \begin{bmatrix} 5 & -0.4 \\ -0.4 & 4 \end{bmatrix} \tag{29}$$

Combination 4: high covariance with low correlation

$$\mu_1 = (00), \Sigma_1 = \begin{bmatrix} 5 & 4.5 \\ 4.5 & 5 \end{bmatrix}$$
$$\mu_2 = (-11), \Sigma_2 = \begin{bmatrix} 5 & 3.6 \\ 3.6 & 4 \end{bmatrix}$$
$$\mu_3 = (2-2), \Sigma_2 = \begin{bmatrix} 3 & 2.3 \\ 2.3 & 3 \end{bmatrix} \tag{30}$$

Based on detection accuracy result in TableI, OBK has higher accuracy and OBK is better choice of online segmentation algorithm. Even the observation has relatively large variance, OBK can adapt the model properly. Especially when these two variable has stronger correlation, OBK generates clearly better result. From below (Figure 5-Figure 8), it is clearly showed that OB method is more sensitive for updating

TABLE I
DETECTION ACCURACY OF OBK AND OB ON FOUR DIFFERENT SIMULATE DATA

	low_cov low_corr	low_cov high_corr	high_cov low_corr	high_cov high_corr
oBK	0.9552	0.9995	0.9238	0.9861
oB	0.9414	0.9002	0.9147	0.8839

the "run length" and the estimated "run" length is more unstable. OBK method already includes adaptive online kernel approach, which result steadier and stronger updating "run length".

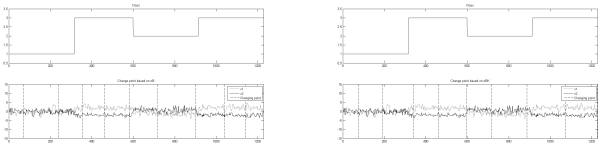

Fig. 5. Detection Accuracy of oB and oKDE on low covariance and low correlation bi-normal simulation data

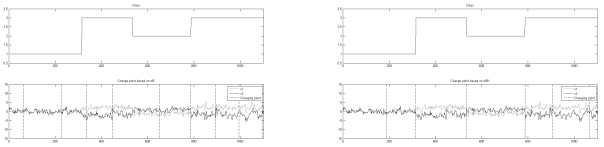

Fig. 6. Detection Accuracy of oB and oKDE on low covariance and low correlation bi-normal simulation data

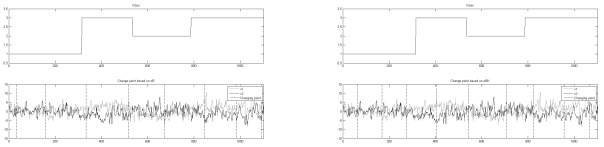

Fig. 7. Detection Accuracy of oB and oKDE on high covariance and low correlation bi-normal simulation data

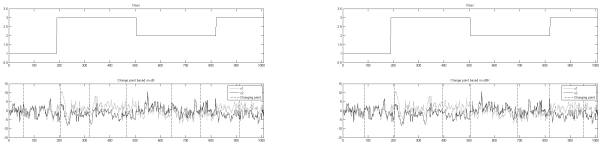

Fig. 8. Detection Accuracy of oB and oKDE on high covariance and high correlation bi-normal simulation data

B. Empirical observation experiment

One of empirical data [31] we used here is three dimensional sensor data collected by cell phone used for tracking human pattern. The data source display six basic activities: standing, sitting, lying, walking, walking downstairs and walking upstairs generated from smartphone that have been carried out with a group of 30 volunteers. The result of using OKDE and OB shows in Table II and Table III.

TABLE II
ONLINE BAYESIAN KERNEL SEGMENTATION CONFUSION MATRIX

	Walking	Upstairs	Downstairs	Sitting	Standing	Laying
Walking	1	0	0	0	0	0
Upstairs	0	0.8895	0.1079	0	0.0026	0
Downstairs	0.0172	0.0989	0.8839	0	0	0
Sitting	0	0	0	0.8941	0.1059	0
Standing	0	0.0445	0	0.1096	0.8450	0.0009
Laying	0	0	0	0.0006	0.0043	0.9993

TABLE III
ONLINE BAYESIAN SEGMENTATION CONFUSION MATRIX

	Walking	Upstairs	Downstairs	Sitting	Standing	Laying
Walking	0.9434	0	0.0053	0	0	0.0053
Upstairs	0.0310	0.7451	0.1720	0	0.0518	0
Downstairs	0.1379	0.3162	0.5364	0	0.0095	0
Sitting	0	0	0	0.7717	0.1080	0.1203
Standing	0	0.0314	0	0.0828	0.8406	0.0452
Laying	0.0196	0	0	0.0557	0.0595	0.8653

The accuracy for each activities is displayed in Table II, which shows this algorithm can automatically and efficiently detect changing and find activities time interval. To compare with OBKS method, we use online Bayesian segmentation algorithm [16] as a optional choice and the confusion table III preforms accuracy rates. It's not easy to distinguish Upstair and Downstair, 31.62 % Downstairs observation are misclassified into Upstair category. OBKS has better performance than OBS for each activities, such as there are 89.4 % Sitting observations correctly classified into Sitting using OBKS and there are 77.17% Sitting observations classified into Sitting using OBS. The overall performance is displayed in Table IV.

TABLE IV
DETECTION ACCURACY OF OB AND OKDE ON FOUR DIFFERENT SIMULATE DATA

	Human Pattern Observations overall
oBK	0.9186
oB	0.7834

IV. CONCLUSION

As a important procedure in data mining, time series segmentation divides a whole time series into disjointed subsequences, each subsequence can be represented as a model, such as distribution and regression model. In addition, the subsequence data sets and their associated models are benefit for other data mining algorithm, such as classification and clustering. We propose a Online Bayesian Kernel Segmentation (OBKS) method that breaks a time series automatically based on Bayesian (OB) method combined with online multivariate kernel density estimation. Instead of considering Bayesian predictive function, we apply online kernel function

as predictive function to get rid of assumption of multi-normality. At this point, normality of observation is not strict requirement. Further, kernel density estimation is more flexible and adaptive that is suitable for any distributions. In this work, we apply this innovated method in simulated data and smart phone accelerometer data. Based on simulated result, OKDE still performance better on four different cases. The overall performance has been showed in Table IV, OBKS also has higher overall accuracy than OB method. In the future work, we can test this method in more applications.

REFERENCES

[1] D. Kulic and Y. Nakamura, "Scaffolding on-line segmentation of full body human motion patterns," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2008, pp. 2860–2866.

[2] F. Lv and R. Nevatia, "Recognition and segmentation of 3-d human action using hmm and multi-class adaboost," in *European conference on computer vision*. Springer, 2006, pp. 359–372.

[3] W. Takano and Y. Nakamura, "Real-time unsupervised segmentation of human whole-body motion and its application to humanoid robot acquisition of motion symbols," *Robotics and Autonomous Systems*, vol. 75, pp. 260–272, 2016.

[4] L. Gillick and S. J. Cox, "Some statistical issues in the comparison of speech recognition algorithms," in *Acoustics, Speech, and Signal Processing, 1989. ICASSP-89., 1989 International Conference on*. IEEE, 1989, pp. 532–535.

[5] J. R. Glass, "A probabilistic framework for segment-based speech recognition," *Computer Speech & Language*, vol. 17, no. 2, pp. 137–152, 2003.

[6] T. Starner and A. Pentland, "Real-time american sign language recognition from video using hidden markov models," in *Motion-Based Recognition*. Springer, 1997, pp. 227–243.

[7] T. Starner, J. Weaver, and A. Pentland, "Real-time american sign language recognition using desk and wearable computer based video," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 12, pp. 1371–1375, 1998.

[8] P. Esling and C. Agon, "Time-series data mining," *ACM Computing Surveys (CSUR)*, vol. 45, no. 1, p. 12, 2012.

[9] T.-c. Fu, "A review on time series data mining," *Engineering Applications of Artificial Intelligence*, vol. 24, no. 1, pp. 164–181, 2011.

[10] E. Keogh, S. Chu, D. Hart, and M. Pazzani, "An online algorithm for segmenting time series," in *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on*. IEEE, 2001, pp. 289–296.

[11] M. Lovrić, M. Milanović, and M. Stamenković, "Algoritmic methods for segmentation of time series: An overview," *Journal of Contemporary Economic and Business Issues*, vol. 1, no. 1, pp. 31–53, 2014.

[12] J. Himberg, K. Korpiaho, H. Mannila, J. Tikanmaki, and H. T. Toivonen, "Time series segmentation for context recognition in mobile devices," in *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on*. IEEE, 2001, pp. 203–210.

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

[14] L. Dobos and J. Abonyi, "Fisher information matrix based time-series segmentation of process data," *Chemical Engineering Science*, vol. 101, pp. 99–108, 2013.

[15] H. Shatkay and S. B. Zdonik, "Approximate queries and representations for large data sequences," in *Data Engineering, 1996. Proceedings of the Twelfth International Conference on*. IEEE, 1996, pp. 536–545.

[16] R. P. Adams and D. J. MacKay, "Bayesian online changepoint detection," *arXiv preprint arXiv:0710.3742*, 2007.

[17] R. Garnett, M. A. Osborne, and S. J. Roberts, "Sequential bayesian prediction in the presence of changepoints," in *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 2009, pp. 345–352.

[18] E. Ruggieri and M. Antonellis, "An exact approach to bayesian sequential change point detection," *Computational Statistics & Data Analysis*, vol. 97, pp. 71–86, 2016.

[19] T. Mori, Y. Nejigane, M. Shimosaka, Y. Segawa, T. Harada, and T. Sato, "Online recognition and segmentation for time-series motion with hmm and conceptual relation of actions," in *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*. IEEE, 2005, pp. 3864–3870.

[20] M. M. Shafiei and H. R. Rabiee, "A new online signature verification algorithm using variable length segmentation and hidden markov models," in *Document Analysis and Recognition, 2003. Proceedings. Seventh International Conference on*. IEEE, 2003, pp. 443–446.

[21] N. Chopin, "Dynamic detection of change points in long time series," *Annals of the Institute of Statistical Mathematics*, vol. 59, no. 2, pp. 349–366, 2007.

[22] M. Kristan, A. Leonardis, and D. Skočaj, "Multivariate online kernel density estimation with gaussian kernels," *Pattern Recognition*, vol. 44, no. 10, pp. 2630–2642, 2011.

[23] M. Kristan and A. Leonardis, "Online discriminative kernel density estimator with gaussian kernels," *IEEE transactions on cybernetics*, vol. 44, no. 3, pp. 355–365, 2014.

[24] C. G. Lambert, S. E. Harrington, C. R. Harvey, and A. Glodjo, "Efficient on-line nonparametric kernel density estimation," *Algorithmica*, vol. 25, no. 1, pp. 37–57, 1999.

[25] S. Na, K. M. Ramachandran, and M. Ji, "Real-time activity recognition using smartphone accelerometer," *Pervasive and Mobile Computing*, 2016 submitted.

[26] P. A. Tobias and D. Trindade, *Applied reliability*. CRC Press, 2011.

[27] M. P. Wand and M. C. Jones, *Kernel smoothing*. Crc Press, 1994.

[28] M. Kristan, D. Skočaj, and A. Leonardis, "Online kernel density estimation for interactive learning," *Image and Vision Computing*, vol. 28, no. 7, pp. 1106–1116, 2010.

[29] S. J. Julier and J. K. Uhlmann, "A general method for approximating nonlinear transformations of probability distributions," Technical report, Robotics Research Group, Department of Engineering Science, University of Oxford, Tech. Rep., 1996.

[30] M. S. Grewal, *Kalman filtering*. Springer, 2011.

[31] J.-L. Reyes-Ortiz, L. Oneto, A. Sama, X. Parra, and D. Anguita, "Transition-aware human activity recognition using smartphones," *Neurocomputing*, vol. 171, pp. 754–767, 2016.