An Incremental Kernel Density Estimator for Data Stream Computation

Yulin He, Jie Jiang, Dexin Dai, and Klohoun Fabrice

College of Computer Science & Software Engineering, Shenzhen University, Shenzhen 518060, China
National Engineering Laboratory for Big Data System Computing Technology, Shenzhen University, Shenzhen 518060, China
Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ), Shenzhen University, Shenzhen 518060, China

Correspondence should be addressed to Yulin He; yulinhe@szu.edu.cn

1.Introduction

Probability density function (p.d.f.) estimation plays a very important role in the field of data mining. Kernel density estimator (KDE) is the mostly used technology to estimate the unknown p.d.f. for the given dataset. The existing KDEs are usually inefficient when handling the p.d.f. estimation problem for stream data because a bran-new KDE has to be retrained based on the combination of current data and newly coming data. This process increases the training time and wastes the computation resource. This article proposes an incremental kernel density estimator (I-KDE) which deals with the p.d.f. estimation problem in the way of data stream computation. The I-KDE updates the current KDE dynamically and gradually with the newly coming data rather than retraining the bran-new KDE with the combination of current data and newly coming data. The theoretical analysis proves the convergence of the I-KDE only if the estimated p.d.f. of newly coming data is convergent to its true p.d.f. In order to guarantee the convergence of the I-KDE, a new multivariate fixed-point iteration algorithm based on the unbiased cross validation (UCV) method is developed to determine the optimal bandwidth of the KDE. The experimental results on 10 univariate and 4 multivariate probability distributions demonstrate the feasibility and effectiveness of the I-KDE.
Learning system in a batch way, the existing KDEs have to retrain the bran-new KDE with the combination of current data and newly coming data. The I-KDE deals with the p.d.f. estimation problem in the way of retraining the bran-new KDE with the combination of current data and newly coming data. The I-KDE updates the p.d.f. estimated with the resampled data to replace the true p.d.f. The biased cross validation (BCV) method [12] is specially designed to estimate the true p.d.f. of error criterion. Assume there are \( T \) (\( T = 1, 2, 3, \ldots \)) datasets \( D_1, D_2, \ldots, D_T \) with \( N_1, N_2, \ldots, N_T \) \( \mathbb{D} \)-dimensional samples, respectively, where

\[
D_t = \{ x_{i}^{(t)} \mid i = 1, 2, \ldots, N_t, d = 1, 2, \ldots, \mathbb{D} \}, \quad t = 1, 2, \ldots, T.
\]

Let

\[
D^{(T)} = \bigcup_{t=1}^{T} D_t = \{ x_{n}^{(T)} \mid x_{n}^{(T)} = (x_{n1}^{(T)}, x_{n2}^{(T)}, \ldots, x_{nD}^{(T)}), \quad x_{nD}^{(T)} \in \mathbb{R},
\]

\[
n = 1, 2, \ldots, N, d = 1, 2, \ldots, \mathbb{D}, \quad \mathbb{N}^{(T)} = \bigcup_{t=1}^{T} \mathbb{N}_t,
\]

represent the combination of datasets \( D_1, D_2, \ldots, D_T \). For any \( x_{n}^{(T)} \in D, n = 1, 2, \ldots, N, \) there exists \( t \in \{ 1, 2, \ldots, T \} \) and \( i \in \{ 1, 2, \ldots, N_t \} \) such that \( x_{n}^{(T)} = z_{i}^{(t)} \). If we want to estimate the underlying p.d.f. for dataset \( D^{(T)} \), the mathematical model of classical KDE [7] is described as

\[
\mathcal{F}_{\mathcal{H}^{(T)}}(x) = \frac{1}{\mathcal{N}^{(T)}} \sum_{n=1}^{\mathcal{N}^{(T)}} \mathcal{K}_{\mathcal{H}^{(T)}}(x - x_{n}^{(T)})
\]

\[
= \frac{1}{\mathcal{N}^{(T)}} \sum_{n=1}^{\mathcal{N}^{(T)}} \left( \left( (x_{1} - x_{n1}^{(T)})/\mathcal{H}_{1}^{(T)} \right), \left( (x_{2} - x_{n2}^{(T)})/\mathcal{H}_{2}^{(T)} \right), \ldots, \left( (x_{\mathbb{D}} - x_{n\mathbb{D}}^{(T)})/\mathcal{H}_{\mathbb{D}}^{(T)} \right) \right) \prod_{d=1}^{\mathbb{D}} \mathcal{H}_{d}^{(T)}
\]

in Section 5.
where
\[
\kappa(x) = \frac{1}{(\sqrt{2\pi})^d} \exp\left(-\frac{1}{2}x^T x\right)
\]
\[
= \prod_{d=1}^D \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}x^2_d\right),
\]
(4)
is the standard \(\mathcal{D}\)-variate Gaussian kernel function and
\[
\Pi^{(\mathcal{D})}(h_1^{(\mathcal{D})}, h_2^{(\mathcal{D})}, \ldots, h_D^{(\mathcal{D})}, h_d^{(\mathcal{D})}) > 0, \quad d = 1, 2, \ldots, \mathcal{D},
\]
is the optimal window bandwidth which can be determined with the different bandwidth selection strategies, e.g., the bootstrap method [10], unbiased cross validation (UCV) method [11], biased cross validation (BCV) method [12], and the solve-the-equation method [13].

In fact, the estimation paradigm as shown in equation (3) has the following three limitations. First, equation (3) cannot retain the information of current p.d.f. when the data arrive at the learning system sequentially. Second, the computation times are obviously different even for the newly coming data with the same size. The earlier the data arrive, the less training time is required. Third, equation (3) does not work well when the data amount is beyond the memory capacity of the computer. The main reason for these limitations is that the existing KDE must put all the data together to estimate a bran-new p.d.f. Our solution mainly focuses on how to retain the current p.d.f. and uses the newly coming data to update the current p.d.f. The research problem of this article is depicted as follows.

\[
\mathcal{F}^{(\mathcal{T}+1)}(x) = \frac{1}{N^{(\mathcal{T}+1)}} \sum_{n=1}^{N^{(\mathcal{T}+1)}} K^{(\mathcal{T}+1)}_{\mathcal{D}}(x - x_n^{(\mathcal{T}+1)}),
\]
\[
\mathcal{F}^{(\mathcal{T}+1)}_{\Pi^{(\mathcal{T}+1)}}(x) = \frac{1}{N^{(\mathcal{T}+1)}} \sum_{n=1}^{N^{(\mathcal{T}+1)}} K^{(\mathcal{T}+1)}_{\Pi^{(\mathcal{T}+1)}}(x - x_n^{(\mathcal{T}+1)}),
\]
\[
\mathcal{F}^{(\mathcal{T}+1)}_{\Pi^{(\mathcal{T}+1)}}(x) = \frac{1}{N^{(\mathcal{T}+1)}} \sum_{n=1}^{N^{(\mathcal{T}+1)}} K^{(\mathcal{T}+1)}_{\Pi^{(\mathcal{T}+1)}}(x - x_n^{(\mathcal{T}+1)}).
\]

Problem Statement. Assume the current p.d.f. estimated by the current learning system based on data \(D^{(\mathcal{T})} = \bigcup_{i=1}^{T} D_i\) is \(\mathcal{F}^{(\mathcal{T})}(x)\), where the positive integer \(\mathcal{T} \geq 1\) is the number of data batches. Our objective is to use the p.d.f. \(\mathcal{F}^{(\mathcal{T}+1)}_{\Pi^{(\mathcal{T}+1)}}(x)\) estimated with newly coming data \(D_{\mathcal{T}+1}\) to update \(\mathcal{F}^{(\mathcal{T})}(x)\) and then obtain the estimated p.d.f. \(\mathcal{F}^{(\mathcal{T}+1)}_{\Pi^{(\mathcal{T}+1)}}(x)\) for data \(D^{(\mathcal{T}+1)} = D^{(\mathcal{T})} \cup D_{\mathcal{T}+1} = \bigcup_{i=1}^{\mathcal{T}+1} D_i\), i.e.,
\[
\mathcal{F}^{(\mathcal{T}+1)}_{\Pi^{(\mathcal{T}+1)}}(x) = \mathcal{F}^{(\mathcal{T})}(x) \oplus \mathcal{F}^{(\mathcal{T}+1)}_{\Pi^{(\mathcal{T}+1)}}(x),
\]
where the mathematical symbol \(\oplus\) represents the incremental updating to previous result.

3. I-KDE: An Incremental Kernel Density Estimator

This section presents an incremental KDE (I-KDE) for the stream data mining. Assume the learning system has received \(\mathcal{T}(\mathcal{T} \geq 1)\) batches of data \(D_1, D_2, \ldots, D_{\mathcal{T}}\) and the newly coming dataset is
\[
D_{\mathcal{T}+1} = \left\{ z_i^{(\mathcal{T}+1)} \middle| z_i^{(\mathcal{T}+1)} = (z_{i,1}^{(\mathcal{T}+1)}, z_{i,2}^{(\mathcal{T}+1)}, \ldots, z_{i,d}^{(\mathcal{T}+1)}) \right\},
\]
\[
z_{id}^{(\mathcal{T}+1)} \in \mathbb{R}, \quad i = 1, 2, \ldots, N_{\mathcal{T}+1}, \quad d = 1, 2, \ldots, \mathcal{D}.
\]

The estimated p.d.f.s \(\mathcal{F}^{(\mathcal{T})}_{\Pi^{(\mathcal{T})}}(x)\), \(\mathcal{F}^{(\mathcal{T}+1)}_{\Pi^{(\mathcal{T}+1)}}(x)\), and \(\mathcal{F}^{(\mathcal{T}+1)}_{\Pi^{(\mathcal{T}+1)}}(x)\) with data \(D^{(\mathcal{T})} = \bigcup_{i=1}^{\mathcal{T}} D_i\), \(D_{\mathcal{T}+1}\), and \(D^{(\mathcal{T}+1)} = D^{(\mathcal{T})} \cup D_{\mathcal{T}+1}\) are
respectively, where 
\[
\hat{H}^{(2)} = \left( h_1^{(2)}, h_2^{(2)}, \ldots, h_{\mathcal{D}}^{(2)} \right), h_d^{(2)} > 0, \quad d = 1, 2, \ldots, \mathcal{D},
\]

is the optimal bandwidth of p.d.f. \( \mathcal{P}^{(2)}(x) \) estimated with the newly coming data \( D_{T+1} \). In equation (10), we can find that it consumes less time to determine the bandwidth \( \hat{H}^{(2)} \) than to determine \( \hat{H}^{(1)} \) and \( \hat{H}^{(2)} \) when \( N^{(2)} \gg N_{T+1} \) and \( N^{(2)} \gg N_{T+1} \). Thus, the I-KDE only considers how to determine the optimal bandwidth for the newly coming data \( D_{T+1} \) and gives up to calculate the bandwidths for both the current data \( D^{(T)} \) and the combination \( D^{(T+1)} \) of current data and newly coming data. Then, Eq. (9) can be represented as 
\[
\mathcal{P}^{(T)}(x) = \mathcal{P}^{(T)}(x) + \mathcal{P}^{(T+1)}(x),
\]

According to equation (11), we can iteratively derive the following equations:
\[
\begin{align*}
\mathcal{P}^{(1)}(x) &= \mathcal{P}^{(1)}(x), \\
\mathcal{P}^{(T-1)}(x) &= \frac{N^{(T-1)}}{N^{(T-1)} + N_T} \mathcal{P}^{(T-1)}(x) + \frac{N_T}{N^{(T-1)} + N_T} \mathcal{P}^{(T)}(x), \\
\mathcal{P}^{(T-2)}(x) &= \frac{N^{(T-2)}}{N^{(T-2)} + N_T} \mathcal{P}^{(T-2)}(x) + \frac{N_T}{N^{(T-2)} + N_T} \mathcal{P}^{(T-1)}(x), \\
& \vdots \\
\mathcal{P}^{(2)}(x) &= \frac{N^{(1)}}{N^{(1)} + N_2} \mathcal{P}^{(1)}(x) + \frac{N_2}{N^{(1)} + N_2} \mathcal{P}^{(2)}(x),
\end{align*}
\]

The learning system only receives the data \( D_1 \), 
\[
\mathcal{P}^{(1)}(x) = \mathcal{P}^{(1)}(x),
\]
holds for \( D^{(1)} = D_1 \). Substituting equation (12) into equation (11) yields the mathematical model of the I-KDE as 
\[
\begin{align*}
\mathcal{P}^{(T+1)}(x) &= \frac{N_1}{\sum_{t=1}^{T} N_t + N_{T+1}} \mathcal{P}^{(1)}(x) \\
&\quad + \frac{N_2}{\sum_{t=1}^{T} N_t + N_{T+1}} \mathcal{P}^{(2)}(x) + \cdots \\
&\quad + \frac{N_T}{\sum_{t=1}^{T} N_t + N_{T+1}} \mathcal{P}^{(T)}(x) \\
&\quad + \frac{N_{T+1}}{\sum_{t=1}^{T} N_t + N_{T+1}} \mathcal{P}^{(T+1)}(x) \\
&= \sum_{k=1}^{T} \left[ \frac{N_k}{\sum_{t=1}^{T} N_t + N_{T+1}} \mathcal{P}^{(k)}(x) \right],
\end{align*}
\]

where \( \hat{H}^{(k)}(k = 1, 2, \ldots, T + 1) \) is the optimal bandwidth of p.d.f. estimated with data \( D^{(k)} \). Equation (14) reveals that the KDE trained based on the union of different data blocks can be decomposed into the different KDEs which are trained based on the corresponding data blocks in an independent way. In equation (14), we can find that the I-KDE is an asymptotic integration model of a series of KDEs which are trained gradually based on the different data blocks. The I-KDE provides an effective way to deal with the p.d.f. estimation problem for stream data and meanwhile makes it possible to estimate the p.d.f. for large-scale data by partitioning it into different data blocks. Figure 1 depicts the procedure of the I-KDE.

We provide a brief analysis to the computation complexity of the I-KDE. The classical unbiased cross validation (UCV) method [11] is used in this article to determine the optimal bandwidth. Its computation complexity is \( O\left( N^2 D \right) \), where \( N \) is the number of samples and \( D \) is the number of sample dimensions. Assume the learning system receives \( T \) batches of data \( D_1, D_2, \ldots, D_T \); the time complexity of the I-KDE is 
\[
T_{\text{I-KDE}} = O\left( N^2 D \right) + O\left( N^2 D \right) + \cdots + O\left( N^2 D \right)
\]

If we reestablish a bran-new p.d.f. for each batch of data by training the KDE based on the combination of current data and newly coming data, the time complexity of full retraining scheme is 
\[
T_{\text{Retraining1}} = O\left( N_1 D \right) + O\left( N_2 D \right) + \cdots + O\left( N_2 D \right)
\]

where
Proof. The probability distribution function $F(x)$ is

$$F(x) = \sum_{i=1}^{T} \frac{N_i}{N} F^{(t)}(x).$$

Then, the probability distribution function $\tilde{F}(x)$ is

$$\tilde{F}(x) = \sum_{i=1}^{T} N_i \tilde{F}^{(t)}(x).$$

This concludes the proof.

\begin{theorem}[convergence of I-KDE] Assume $D_t$ includes $N_t$ observations of $D$-dimensional random variables $X_t$ of which the p.d.f. is $\varphi^{(t)}(x)$, $t = 1, 2, \ldots, T$ and $D = \bigcup_{t=1}^{T} D_t$. If $X_1, X_2, \ldots, X_T$ are mutually independent and there exists $\varepsilon > 0$ such that $|\varphi^{(t)}(x) - \tilde{\varphi}^{(t)}(x)| < \varepsilon$ for $t = 1, 2, \ldots, T$, then

$$|\varphi(x) - \tilde{\varphi}(x)| < \sum_{i=1}^{T} N_i \varepsilon_t,$$

where $\tilde{\varphi}^{(t)}(x)$ is the estimated p.d.f. with the KDE on data $D_t$ and $\varphi(x) = \sum_{i=1}^{T} N_i \varphi^{(t)}(x)$ are the estimated p.d.f.s with the I-KDE on data $D$.

Proof. Based on Lemma 1, we have

$$\tilde{\varphi}(x) = \sum_{i=1}^{T} \frac{N_i}{N} \varphi^{(t)}(x).$$

Then,

$$|\varphi(x) - \tilde{\varphi}(x)| \leq \sum_{i=1}^{T} \frac{N_i}{N} |\varphi^{(t)}(x) - \tilde{\varphi}^{(t)}(x)|.$$

For the given $\varepsilon_t > 0, t = 1, 2, \ldots, T, |\varphi^{(t)}(x) - \tilde{\varphi}^{(t)}(x)| < \varepsilon_t$ holds. Thus, we can derive

$$|\varphi(x) - \tilde{\varphi}(x)| \leq \sum_{i=1}^{T} \frac{N_i}{N} |\varphi^{(t)}(x) - \tilde{\varphi}^{(t)}(x)| \leq \sum_{i=1}^{T} N_i \varepsilon_t,$$

This completes the proof.

Note. When $\varepsilon_t \rightarrow 0$ for $t = 1, 2, \ldots, T$, then $\sum_{i=1}^{T} (N_i/\varepsilon_t) \rightarrow 0$. It means that $\tilde{\varphi}(x)$ can approximate $\varphi(x)$ well if $\tilde{\varphi}^{(t)}(x)$ can approximate $\varphi^{(t)}(x)$ well.

Theorem 1 demonstrates the convergence of the I-KDE when the estimated p.d.f. $\tilde{\varphi}^{(t)}(x)$ converges to the true p.d.f. $\varphi^{(t)}(x)$. In order to obtain an accurate p.d.f. estimation for
In equation (14), we use the multivariate fixed-point iteration to design the bandwidth optimization method based on UCV error criterion. The formulation [12] of UCV error criterion for $D$-dimensional observations

$$D_t = \{ z^{(t)}_i \, | \, z_i^{(t)} = (z_{i1}^{(t)}, z_{i2}^{(t)}, \ldots, z_{id}^{(t)}) \in \mathbb{R}, i = 1, 2, \ldots, N_t, d = 1, 2, \ldots, D \}$$

is

$$UCV(H^{(t)}) = UCV(h_1^{(t)}, h_2^{(t)}, \ldots, h_D^{(t)})$$

$$= \frac{1}{(2\sqrt{\pi})^D N_t^2 h_1^{(t)} h_2^{(t)} \ldots h_D^{(t)}}$$

$$+ \frac{1}{(2\sqrt{\pi})^D N_t^2 h_1^{(t)} h_2^{(t)} \ldots h_D^{(t)}}$$

$$\times \sum_{i=1}^{N_t} \sum_{j=1}^{N_t} \left[ \exp \left[ -\frac{1}{4} \sum_{d=1}^{D} \left( \Delta_{ijd}^{(t)} \right)^2 \right] \right],$$

where

$$\Delta_{ijd}^{(t)} = \frac{z_{id}^{(t)} - z_{jd}^{(t)}}{h_d^{(t)}}. \quad (29)$$

The optimal bandwidth for the estimated p.d.f. $\mathcal{D}^{(t)}(x)$ satisfies

$$H^{(t)} = \arg \min_{h_d^{(t)} > 0, d = 1, 2, \ldots, D} UCV(H^{(t)}). \quad (30)$$
We calculate the partial derivative of $UCV(H^{(t)})$ with respect to $h_d^{(t)}$, $d = 1, 2, \ldots, D$ and let

$$\frac{\partial UCV(H^{(t)})}{\partial h_d^{(t)}} = -\left(\frac{1}{(2\sqrt{\pi})^D A_t} \prod_{k=1}^{D} h_k^{(t)} \right) \left[ h_d^{(t)} \right]^2$$

$$+ \left(\frac{1}{(2\sqrt{\pi})^D A_t} \prod_{k=1}^{D} h_k^{(t)} \right) \left[ h_d^{(t)} \right]^2 - \Delta_{d1}^{(t)}(h_1^{(t)}, h_2^{(t)}, \ldots, h_D^{(t)})$$

$$+ \left(\frac{1}{(2\sqrt{\pi})^D A_t^2} \prod_{k=1}^{D} h_k^{(t)} \right) \left[ h_d^{(t)} \right]^4 - \Delta_{d2}^{(t)}(h_1^{(t)}, h_2^{(t)}, \ldots, h_D^{(t)}) = 0$$

where

$$\Delta_{d1}^{(t)}(h_1^{(t)}, h_2^{(t)}, \ldots, h_D^{(t)}) = \sum_{i=1}^{X_t} \sum_{j=1}^{X_t} \exp \left[ -\frac{1}{4} \sum_{d=1}^{D} \left[ \Delta_{ijd}^{(t)} \right]^2 \right] - (2 \times 2^{D/2}) \exp \left[ -\frac{1}{2} \sum_{d=1}^{D} \left[ \Delta_{ijd}^{(t)} \right]^2 \right]$$

$$\Delta_{d2}^{(t)}(h_1^{(t)}, h_2^{(t)}, \ldots, h_D^{(t)}) = \sum_{i=1}^{X_t} \sum_{j=1}^{X_t} \left[ \frac{1}{2} z_{id}^{(t)} - z_{jd}^{(t)} \right]^2 \exp \left[ -\frac{1}{4} \sum_{d=1}^{D} \left[ \Delta_{ijd}^{(t)} \right]^2 \right] - (2 \times 2^{D/2}) \left[ \frac{1}{2} z_{id}^{(t)} - z_{jd}^{(t)} \right]^2 \exp \left[ -\frac{1}{2} \sum_{d=1}^{D} \left[ \Delta_{ijd}^{(t)} \right]^2 \right].$$

It is very difficult to calculate the analytic solution of $h_d^{(t)}$, $d = 1, 2, \ldots, D$ from equation (31). However, an iterative function with respect to $h_d^{(t)}$ can be derived by simplifying equation (31):

$$h_d^{(t)} = \frac{\Delta_{d1}^{(t)}(h_1^{(t)}, h_2^{(t)}, \ldots, h_D^{(t)})}{\Delta_{d1}^{(t)}(h_1^{(t)}, h_2^{(t)}, \ldots, h_D^{(t)}) + \Delta_{d2}^{(t)}(h_1^{(t)}, h_2^{(t)}, \ldots, h_D^{(t)})}$$

Furthermore, a multivariate fixed-point iteration algorithm can be designed based on the aforementioned iterative function to determine the optimal bandwidth $H^{(t)} = (h_1^{(t)}, h_2^{(t)}, \ldots, h_D^{(t)})$.

4. Experimental Results and Analysis

In this section, we conduct three experiments to demonstrate the feasibility and effectiveness of our proposed I-KDE based on 10 univariate and 4 multivariate probability distributions of which the details are listed in Table 1 [19]. First, we validate the convergence of Algorithm 1. Second, we demonstrate the convergence of the I-KDE. Third, we test the p.d.f. estimation performance of the I-KDE.

4.1. Convergence of Algorithm 1. In this experiment, we check the convergence of Algorithm 1 on 2 univariate (beta and normal) and 2 multivariate (2-dimensional bimodal and
Table 1: The details of 10 univariate and 4 multivariate probability distributions \( (\mathcal{P}(i)(x)) \) in bimodal, trimodal, and quadrimodal normal distributions is the \( D \)-dimensional normal distribution with mean vector \( \Theta(i) = (\mu_1(i), \mu_2(i), \ldots, \mu_D(i)) \) and covariance matrix \( \Sigma(i) = \begin{bmatrix} 
\sigma_{11}(i) & \sigma_{12}(i) & \cdots & \sigma_{1D}(i) \\
\sigma_{21}(i) & \sigma_{22}(i) & \cdots & \sigma_{2D}(i) \\
\vdots & \vdots & \ddots & \vdots \\
\sigma_{D1}(i) & \sigma_{D2}(i) & \cdots & \sigma_{DD}(i) 
\end{bmatrix} \).

| # | Probability distribution | p.d.f. |
|---|-------------------------|--------|
| 1 | Beta                    | \( \mathcal{P}(x) = (x^\alpha - 1)^{\beta - 1}/(\Gamma(\alpha, \beta)) \), \( x \in (0, 1) \) |
| 2 | Chi-squared             | \( \mathcal{P}(x) = (2 - \lambda x^k e^{\lambda})/k! \), \( x \in (0, 1) \) |
| 3 | Exponential             | \( \mathcal{P}(x) = (0.5e^{-x^2})/\sqrt{\pi} \), \( x \in (0, 1) \) |
| 4 | F                       | \( \mathcal{P}(x) = (2\pi x^{3/2} - 1)/(\Gamma(3/2)) \), \( x \in (0, 1) \) |
| 5 | Gamma                   | \( \mathcal{P}(x) = (1/\sqrt{2\pi \sigma^2})e^{-(x-\mu)^2/(2\sigma^2)} \), \( x \in (0, 1) \) |
| 6 | Lognormal               | \( \mathcal{P}(x) = N(\mu, \sigma) = (1/\sqrt{2\pi \sigma^2}) e^{-(x-\mu)^2/(2\sigma^2)} \), \( x \in (0, 1) \) |
| 7 | Normal                  | \( \mathcal{P}(x) = (\pi \sigma^2)^{-1/2}e^{-x^2/(2\sigma^2)} \), \( x \in (0, 1) \) |
| 8 | Rayleigh                | \( \mathcal{P}(x) = (\pi \sigma^2)^{-1/2}e^{-x^2/(2\sigma^2)} \), \( x \in (0, 1) \) |
| 9 | Student’s T             | \( \mathcal{P}(x) = (1/(\sqrt{(\pi \sigma^2)^2} + x^2)) \), \( x \in (0, 1) \) |
| 10| Weibull                 | \( \mathcal{P}(x) = (\beta/\sigma^2)^{\alpha/(\alpha+1)}e^{-x/(\alpha+1)} \), \( x \in (0, 1) \) |

For univariate normal \( D = 1 \) and multivariate normal \( D = 2 \) – 4.

| # | Probability distribution | p.d.f. |
|---|-------------------------|--------|
| 1 | \( D \)-dimensional standard normal | \( \mathcal{P}(x) = (2\pi)^{-D/2}|\Sigma|^{-1/2}e^{-(x-\Theta)^T \Sigma^{-1} (x-\Theta)} \), \( x = (x_1, x_2, \ldots, x_D) \) |
| 2 | \( D \)-dimensional bimodal normal | \( \mathcal{P}(x) = \sum_{i=1}^{2} \mathcal{P}(i)(x), \sum_{i=1}^{2} \epsilon_i = 1, \epsilon_i \geq 0, i = 1, 2 \) |
| 3 | \( D \)-dimensional trimodal normal | \( \mathcal{P}(x) = \sum_{i=1}^{3} \mathcal{P}(i)(x), \sum_{i=1}^{3} \epsilon_i = 1, \epsilon_i \geq 0, i = 1, 2, 3 \) |
| 4 | \( D \)-dimensional quadrimodal normal | \( \mathcal{P}(x) = \sum_{i=1}^{4} \mathcal{P}(i)(x), \sum_{i=1}^{4} \epsilon_i = 1, \epsilon_i \geq 0, i = 1, 2, 3, 4 \) |
quadrimodal normal) probability distributions. For each distribution, 1000 samples are randomly generated under Matlab programming environment. The experimental results are listed in Figure 3 and confirm that our designed fixed-point iteration algorithm can find the optimal bandwidths for univariate and multivariate UCV error criteria.

For the univariate probability distribution, e.g., the beta distribution in Figure 3(a), we can see that the UCV error first decreases and then increases with the increase of bandwidth parameter. For the any initial bandwidth value 0.001 or 1, our designed fixed-point iteration algorithm can find the optimal bandwidth 0.048 for the I-KDE. For the multivariate case, we use 10 different probability distributions as shown in Table 1. For each distribution, 5000 training samples are randomly generated and 2000 testing samples are used to check the ratio of testing samples with the smaller absolute error to all testing samples. In Figure 4, we can see that the JS divergences are small and the quantitative measures also reflect that the differences between KDEs and I-KDEs are small.

We also test the convergence performance of the I-KDE on the testing data corresponding to beta (α = 2 and β = 2) and 2-dimensional trimodal normal (Θ(1) = (−5, −5), Θ(2) = (5, −3), Θ(3) = (−2, 5)Σ(1) = Σ(2) = Σ(3) = [0.5, 0, 0, 0.5] and ε1 = ε2 = ε3 = ε4 = 1/4 for D5). We can see that the I-KDE and KDE obtain the similar probability distributions with the different estimation ways. The experimental results visually reflect that the I-KDE can converge to the KDE in an incremental manner.

In addition, we use the Jensen–Shannon (JS) divergence to measure the similarity between two different p.d.f.s estimated with the KDE and I-KDE, respectively. The JS divergence is a variant of Kullback–Leibler (KL) divergence [20]. The smaller the JS divergence is, the more similar the two p.d.f.s are. In Figure 4, we can see that the JS divergences are small and the quantitative measures also reflect that the differences between KDEs and I-KDEs are small.

4.2. Convergence of I-KDE. The objective of this experiment is to check whether the estimation of I-KDE which is trained by means of data stream computation can converge to the estimation of KDE which is trained based on the combination of current data and newly coming data. The initial bandwidths in Algorithm 1 are determined based on the RoT method. The experimental results are presented in Figure 4.

For the univariate case, we use 10 different probability distributions as shown in Table 1. For each distribution, 5000 samples are randomly generated. The parameter settings of these p.d.f.s are summarized as follows: α = 2 and β = 2 for beta, k = 2 for chi-squared, μ = 1.5 for exponential, n1 = 100 and n2 = 100 for F, α = 9 and β = 0.5 for gamma, μ = 0 and σ = 0.25 for lognormal, μ = 0 and α = 1 for normal, β = 1 for Rayleigh, ν = 1 for Student’s T, and α = 1 and β = 5 for Weibull. Figure 4(a), we can see that the red and green curves almost coincide with the coming of new data. It is very hard to distinguish the red and green curves.

The similar situation exists in the multivariate case. For the multivariate case, 5 different probability distributions are selected. In Figure 4(b), the parameter settings of these p.d.f.s are summarized as follows: Θ = (2.5, 2.5) and Σ = [1, 0; 0, 1] for D1, Θ = (0, 0) and Σ = [1, 1.5; 1.5, 3] for D2, Θ(1) = (−2.5, −2.5), Θ(2) = (2.5, 2.5), Σ(1) = Σ(2) = [1, 0; 0, 1] and ε1 = ε2 = ε3 = ε4 = 1/2 for D3, Θ(1) = (−2.5, −2.5), Θ(2) = (2.5, 2.5), Θ(3) = (0, 0), Σ(1) = Σ(2) = Σ(3) = [1, 0; 0, 1] for D4, and Σ = [0.5, 0, 0.5; 0.5, 0, 0.5] and ε1 = ε2 = ε3 = ε4 = 1/2 for D5. In Figure 4(a), the black points are training samples and the red points are the testing samples estimated by the I-KDE with the smaller absolute error. We can see that the estimation performance of the I-KDE will gradually converge to a stable state with the increase of training samples.

4.3. Estimation Performance of I-KDE. Table 2 presents the comparison between the KDE and I-KDE on 10 univariate and 4 multivariate distributions, where the parameter

---

Algorithm 1: A multivariate fixed-point iteration algorithm to determine the optimal bandwidth for the I-KDE.
Figure 3: Continued.
**Figure 3:** The convergence of Algorithm 1 on 2 univariate and 2 multivariate distributions (a) Beta distribution ($\alpha = 2, \beta = 5$, and $h = 0.048$). (b) Normal distribution ($\mu = 0, \sigma = 1$, and $h = 0.331$). (c) 2-dimensional Bimodal normal distribution ($\Theta^{(1)} = (-2.5, -2.5)$, $\Sigma^{(1)}$ and $\Sigma^{(2)}$ are the identity matrices $\epsilon_1 = \epsilon_2 = 1/2, h_1 = 0.546$ and $h_2 = 0.595$). (d) 2-dimensional Quadrimodal normal distribution ($\Theta^{(1)} = (-2.5, -2.5)$, $\Theta^{(2)} = (2.5, 2.5), \Theta^{(3)} = (-2.5, 2.5)$, $\Theta^{(4)} = (2.5, -2.5)$, $\Sigma^{(1)}, \Sigma^{(2)}, \Sigma^{(3)}$, and $\Sigma^{(4)}$ are the identity matrices $\epsilon_1 = \epsilon_2 = \epsilon_3 = \epsilon_4 = 1/4, h_1 = 0.451$ and $h_2 = 0.575$).
Figure 4: Continued.
settings of beta, chi-squared, exponential, F, gamma, log-normal, Rayleigh, Student’s T, Weibull, 2-dimensional bimodal normal, and 2-dimensional trimodal normal distributions are the same as Figures 4(a), 5, and 6. The p.d.f. of Bimodal normal is \( \mathcal{P}(x) = 3/4N(0, 1) + 1/4N(3/2, 1/3) \). The parameters of 2-dimensional standard normal distribution are \( \Theta = (0, 0) \) and \( \Sigma = [1, 0; 0, 1] \). The parameters of 2-dimensional quadrimodal normal distributions are \( \Theta^{(1)} = (-2, -4), \Theta^{(2)} = (3, 4), \Theta^{(3)} = (-4, 3), \Theta^{(4)} = (4, -2), \Sigma^{(1)} = \Sigma^{(2)} = \Sigma^{(3)} = \Sigma^{(4)} = [0.5, 0; 0, 0.5] \), and \( \varepsilon_1 = \varepsilon_2 = \varepsilon_3 = \varepsilon_4 = 1/4 \). In addition, the initial bandwidths in Algorithm 1 are determined based on the RoT method. For each distribution, the mean absolute error (MAE) and number of win samples are the average values of 10 runs of different trainings. We give a statistical analysis to the experimental results in Table 2 and confirm that the I-KDE is statistically convergent to the KDE. Taking Exponential distribution for example, we use the Wilcoxon signed-ranks test [21] to check whether the difference between the I-KDE and KDE is significant. Table 3 lists the comparative results of the I-KDE and KDE on 10 different training datasets of exponential distribution.

Let \( R^+ \) be the sum of ranks for the training dataset on which the KDE outperforms the I-KDE and \( R^- \) be the sum of ranks for the training datasets on which the I-KDE outperforms the KDE. We can calculate

\[
R^+ = \sum_{d_i > 0} \text{rank}(d_i) + \frac{1}{2} \sum_{d_i = 0} \text{rank}(d_i) = 41 \tag{34}
\]
Figure 5: The convergence of the I-KDE on beta and 2-dimensional trimodal normal distributions. (a) Univariate case on beta distribution (99 testing samples are selected from 0.01 to 0.99 with incremental step of 0.01). (b) Multivariate case on 2-dimensional trimodal normal distribution (40000 testing samples are selected with the incremental steps of (0.01, 0.01) in the space of $[-9.9, 10] \cup [-9.9, 10]$).
Figure 6: The convergence tendency of the I-KDE on 2-dimensional bimodal normal distribution (the value in the parenthesis is the win ratio of the I-KDE). (a) 1 data block (1.000). (b) 2 data blocks (0.783). (c) 3 data blocks (0.790). (d) 4 data blocks (0.739). (e) 5 data blocks (0.748). (f) 6 data blocks (0.727). (g) 7 data blocks (0.705). (h) 8 data blocks (0.675). (i) 9 data blocks (0.672). (j) 10 data blocks (0.613). (k) 11 data blocks (0.625). (l) 12 data blocks (0.626). (m) 13 data blocks (0.634). (n) 14 data blocks (0.628). (o) 15 data blocks (0.621). (p) 16 data blocks (0.638). (q) 17 data blocks (0.620). (r) 18 data blocks (0.614). (s) 19 data blocks (0.610). (t) 20 data blocks (0.604).

Table 2: The comparison between the KDE and I-KDE on 10 univariate and 4 multivariate distributions.

| # | Probability distribution | Training samples | Testing space | Testing samples | KDE MAE | Win samples | I-KDE MAE | Win samples |
|---|--------------------------|------------------|---------------|----------------|---------|------------|-----------|------------|
| 1 | Beta                     | 2000             | [0.01, 0.99]  | 99             | 0.053   | 52         | 0.048     | 47         |
| 2 | Chi-squared              | 2000             | [0.01, 20]   | 2000          | 0.005   | 437        | 0.004     | 1563       |
| 3 | Exponential              | 2000             | [0.01, 10]   | 1000          | 0.014   | 178        | 0.013     | 822        |
| 4 | F                        | 2000             | [0.01, 10]   | 1000          | 0.007   | 241        | 0.008     | 759        |
| 5 | Gamma                    | 2000             | [1.01, 11]   | 1000          | 0.005   | 540        | 0.008     | 460        |
| 6 | Lognormal                | 2000             | [0.01, 20]   | 2000          | 0.005   | 501        | 0.006     | 1499       |
| 7 | Bimodal normal           | 2000             | [−4.99, 5]   | 1000          | 0.040   | 252        | 0.038     | 748        |
| 8 | Rayleigh                 | 2000             | [0.01, 1]    | 100           | 0.062   | 45         | 0.064     | 55         |
| 9 | Student’s $T$            | 2000             | [−9.99, 10]  | 2000          | 0.003   | 820        | 0.004     | 1180       |
| 10| Weibull                  | 2000             | [0.01, 10]   | 1000          | 0.012   | 176        | 0.010     | 824        |
In this article, we proposed an incremental kernel density estimator (I-KDE) to deal with the probability density estimation problem of large-scale data in the way of data stream computation. In future, we will try to combine the I-KDE with the random sample partition (RSP) model [22, 23] of big data and seek the practical applications for the I-KDE, e.g., Bayesian classification, density-based clustering, and big data reduction [24].

### 5. Conclusions and Future Works

In this article, we proposed an incremental kernel density estimator (I-KDE) to deal with the probability density function (p.d.f.) estimation problem in the way of data stream computation. The I-KDE updated the current KDE dynamically and gradually with the newly coming data rather than retraining the bran-new KDE with the combination of current data and newly coming data. The theoretical analysis proved the convergence of the I-KDE only if the estimated p.d.f. of newly coming data is convergent to its true p.d.f. The experimental results on 10 univariate and 4 multivariate probability distributions demonstrated that the I-KDE obtained the equivalent p.d.f. estimation performance with the less training time in comparison to the KDE and thus indicated that the I-KDE can be used to deal with the p.d.f. estimation problem of large-scale data in the way of data stream computation. In future, we will try to combine the I-KDE with the random sample partition (RSP) model [22, 23] of big data and seek the practical applications for the I-KDE, e.g., Bayesian classification, density-based clustering, and big data reduction [24].

### Data Availability

The data used in our manuscript can be accessed by readers via our BaiduPan at https://pan.baidu.com/s/1wgj-gTzeEZL51WTh2RpCzA with the extraction code “kai5”.

### Conflicts of Interest

The authors declare that they have no conflicts of interest.
Acknowledgments

The authors thank Dr. Salman Salloum (ssalloum@szu.edu.cn) for his language quality improvement and Dr. Xiaoliang Zhang (zhangxla@163.com) for his mathematical derivations. This study was supported by the National Key R&D Program of China (2017YFC0822604-2), Scientific Research Foundation of Shenzhen University for Newly Introduced Teachers (2018060), and National Training Program of Innovation and Entrepreneurship for Undergraduates (201910590017).

References

[1] B. W. Silverman, Density Estimation for Statistics and Data Analysis, CRC Press, Boca Raton, FL, USA, 1986.

[2] G. H. John and P. Langley, “Estimating continuous distributions in Bayesian classifiers,” in Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence, pp. 338–345, Montreal, Canada, August 1995.

[3] A. Azzalini and N. Torelli, “Clustering via nonparametric density estimation,” Statistics and Computing, vol. 17, no. 1, pp. 71–80, 2007.

[4] H. C. Peng, F. Long, and C. Ding, “Feature selection based on mutual information criteria of max-dependency, max-relevancy, and min-redundancy,” IEEE Transactions on Pattern Analysis & Machine Intelligence, vol. 27, no. 8, pp. 1226–1238, 2005.

[5] A. Harvey and V. Oryshchenko, “Kernel density estimation for time series data,” International Journal of Forecasting, vol. 28, no. 1, pp. 3–14, 2012.

[6] B. Poczos, L. Xiong, D. J. Sutherland, and J. Schneider, “Nonparametric kernel estimators for image classification,” in Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp. 2989–2996, Providence, RI, USA, June 2012.

[7] E. Parzen, “On estimation of a probability density function and mode,” The Annals of Mathematical Statistics, vol. 33, no. 3, pp. 1065–1076, 1962.

[8] M. P. Wand and M. C. Jones, Kernel Smoothing, Chapman and Hall, London, UK, 1994.

[9] B. W. Silverman, Density Estimation for Statistics and Data Analysis, Routledge, Abingdon, UK, 2018.

[10] J. J. Faraway and M. Jhun, “Bootstrap choice of bandwidth for density estimation,” Journal of the American Statistical Association, vol. 85, no. 412, pp. 1119–1122, 1990.

[11] A. W. Bowman, “An alternative method of cross-validation for the smoothing of density estimates,” Biometrika, vol. 71, no. 2, pp. 353–360, 1984.

[12] S. R. Sain, K. A. Baggerly, and D. W. Scott, “Cross-validation of multivariate densities,” Journal of the American Statistical Association, vol. 89, no. 427, pp. 807–817, 1994.

[13] L. A. Alexandre, A Solve-The-Equation Approach for Uni-dimensional Data Kernel Bandwidth Selection, 2008, http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.309.4041.

[14] A. Delaigle and I. Gijbels, “Bootstrap bandwidth selection in kernel density estimation from a contaminated sample,” Annals of the Institute of Statistical Mathematics, vol. 56, no. 1, pp. 19–47, 2004.

[15] T. Duong and M. L. Hazelton, “Cross-validation bandwidth matrices for multivariate kernel density estimation,” Scandinavian Journal of Statistics, vol. 32, no. 3, pp. 485–506, 2005.

[16] J. M. Wang, R. U. Yang, Y. Chen, and Y. X. Xu, “Solve-the-equation kernel density estimation method based on cosine kernel function,” Journal of Harbin University of Science and Technology, vol. 21, no. 1, pp. 114–117, 2016.

[17] H. H. Nasereddin, “Stream data mining,” International Journal of Web Applications, vol. 3, no. 2, pp. 90–97, 2011.

[18] H. B. He, S. Chen, K. Li, and X. Xu, “Incremental learning from stream data,” IEEE Transactions on Neural Networks, vol. 22, no. 12, pp. 1901–1914, 2011.

[19] J. Jiang, Y. L. He, D. X. Dai, and J. Z. Huang, “A new kernel density estimator based on the minimum entropy of data set,” Information Sciences, vol. 491, pp. 221–231, 2019.

[20] F. Pérez-Cruz, “Kullback-Leibler divergence estimation of continuous distributions,” in Proceedings of the 2008 IEEE International Symposium on Information Theory, pp. 1666–1670, Toronto, Canada, July 2008.

[21] J. Demšar, “Statistical comparisons of classifiers over multiple data sets,” Journal of Machine Learning Research, vol. 7, pp. 1–30, 2006.

[22] S. Salloum, J. Z. Huang, and Y. He, “Random sample partition: a distributed data model for big data analysis,” IEEE Transactions on Industrial Informatics, vol. 15, no. 11, pp. 5846–5854, 2019.

[23] S. Salloum, J. Z. Huang, Y. L. He, and X. J. Chen, “An asymptotic ensemble learning framework for big data analysis,” IEEE Access, vol. 7, pp. 3675–3693, 2018.

[24] M. H. ur Rehman, C. S. Liew, A. Abbas, P. P. Jayaraman, T. Y. Wah, and S. U. Khan, “Big data reduction methods: a survey,” Data Science and Engineering, vol. 1, no. 4, pp. 265–284, 2016.