Isolation Kernel Density Estimation

Kai Ming Ting
National Key Laboratory for Novel Software Technology, Nanjing University, China
tingkm@nju.edu.cn

Takashi Washio
The Institute of Scientific and Industrial Research, Osaka University, Japan
washio@ar.sanken.osaka-u.ac.jp

Jonathan. R. Wells
School of Information Technology, Deakin University, Australia
jonathan.wells.research@gmail.com

Hang Zhang
National Key Laboratory for Novel Software Technology, Nanjing University, China
zhanghang@lamda.nju.edu.cn

2021 IEEE International Conference on Data Mining (ICDM) | 978-1-6654-2398-4/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICDM51629.2021.00073

***Abstract*—This paper shows that adaptive kernel density estimator (KDE) can be derived effectively from Isolation Kernel. Existing adaptive KDEs often employ a data independent kernel such as Gaussian kernel. Therefore, it requires an additional means to adapt its bandwidth locally in a given dataset. Because Isolation Kernel is a data dependent kernel which is derived directly from data, no additional adaptive operation is required. The resultant estimator called `IKDE` is the only KDE that is fast and adaptive. Existing KDEs are either fast but non-adaptive or adaptive but slow. In addition, using `IKDE` for anomaly detection, we identify two advantages of `IKDE` over LOF (Local Outlier Factor), contributing to significantly faster runtime.**

***Index Terms*—Isolation Kernel, Kernel Density Estimation, Anomaly Detection**

I. Introduction

Kernel density estimator (KDE) is a non-parametric method that estimates the probability density function of a data distribution from a given dataset. KDE is important because it has been proved that virtually all non-parametric density estimation methods are asymptotically kernel methods [1].

Despite its sound theoretical backing, KDE has not been widely used because of its high computational cost. This is due to two issues: (a) each estimation costs $O(n)$ if the dataset used has n points; and (b) the adaptive version of KDE requires additional computational expense to find a bandwidth adaptive to local density. For more than 60 years since the advent of KDE [2], an efficient adaptive KDE has evaded numerous efforts in the research community.

There are at least four current KDE approaches. First, since the introduction of Fast Fourier Transform (FFT) [3], it has been used to improve the time complexity of KDE via a binning method (e.g., [4], [5], [6], [7]). While the FFT-based method has been very successful in univariate KDE, it has difficulty extending to multivariate KDE of more than three dimensions. This is because the number of bins (in a multivariate dataset) grows exponentially with respect to the number of dimensions [8]. Although this issue is one facet of the curse of dimensionality, one can conclude that the current FFT-based methods are prone to the curse that limits their use to few dimensions only.

Second, a kernel functional approximation approach holds promise in achieving the aim of linear time multivariate KDE. A representative is the Nyström method [9]. It creates a finite-dimensional feature map which approximates a feature map of intractable dimensionality of a kernel such as Gaussian kernel.

Third, a recent approach employs locality-sensitive hashing (LSH) [10], [11], [12] to compute the kernel summary of a data independent kernel in order to produce a fast KDE. Like the first and second approaches, they focus on finding efficient ways to approximate the kernel or kernel summary.

Fourth, it aims to produce an adaptive KDE using a data independent kernel [1], [13], by adapting the bandwidth to local density. These KDEs have at least quadratic time complexity.

In a nutshell, existing approaches produce KDEs which are either fast but non-adaptive or slow but adaptive. All of them employ a *data independent* kernel which has a feature map of intractable dimensionality.

Our approach differs from these approaches in one key aspect, i.e., it employs a *data dependent* kernel which has no closed-form expression but a sparse and finite-dimensional feature map called Isolation Kernel (IK) [14]. The insight is that the data dependent IK with a fixed sharpness parameter achieves the same outcome of a data independent kernel with a bandwidth adjusted to local density. We show that the data independent kernel is the key obstacle in producing a fast adaptive KDE. We call the resultant estimator: Isolation Kernel Density Estimator or `IKDE`.

The contributions of this paper are:

1) Introducing an Adaptive Kernel Density Estimator using Isolation Kernel called `IKDE` which is the only KDE that is fast and adaptive. Existing KDEs are either fast or adaptive (but slow).
2) Revealing via a bias-variance analysis that `IKDE` is a consistent estimator under less stringent assumptions than the ordinary KDE based on a Gaussian kernel.
3) Uncovering that Isolation Kernel is the key in producing a fast and adaptive KDE.
4) Examining the efficiency and efficacy of `IKDE` in anomaly detection.
5) Using `IKDE` for anomaly detection, identifying two advantages of `IKDE` over LOF.

Specifically, `IKDE` is distinguished from existing KDEs in the following aspects:

- The isolating partitions used to derive IK is the unique feature of `IKDE`. The isolating partitions have an explicit characteristic: the size of the isolating partition is proportional to the local density of the region. This leads to a unique data dependent kernel: the similarity it measures

978-1-6654-2398-4/21/$31.00 ©2021 IEEE
DOI 10.1109/ICDM51629.2021.00073

depends on the local density (see Lemma 1 in Section III.)

- The adaptive nature of isolating partitions is the key in building an adaptive KDE, without an expensive process to locally adjust bandwidth of a data independent kernel. No existing adaptive KDEs have this ability.
- The sparse feature map, derived from the isolating partitions, contributes directly to $O(1)$ estimation time, after a $O(n)$ preprocessing to compute the kernel mean map. Although some recent LSH-based methods have similar estimation time, they are non-adaptive KDEs. `IKDE` is the only adaptive KDE that can run fast.

We describe existing work in relation to kernel density estimation in the next section, followed by a section on data dependent Isolation Kernel. We present `IKDE` and its analysis to show that `IKDE` is a consistent estimator in Sections IV and V, respectively. Time complexities are provided in Section VI. The empirical evaluation of using KDE for anomaly detection is provided in Section VII. Discussion and conclusions are provided in the last two sections.

II. Kernel density estimation

Given a dataset $Y \subset \mathbb{R}^d$ and a kernel κ_σ with bandwidth parameter σ, KDE for any point $x \in \mathbb{R}^d$ is defined as follows:

$$f_\sigma(x|Y) = \frac{1}{|Y|} \sum_{y \in Y} \kappa_\sigma(x, y) \tag{1}$$

where κ_σ is a non-negative real-valued integrable function satisfying $\int_{\mathbb{R}^d} \kappa_\sigma(x, y)dy = 1$ (normalization); and $\kappa_\sigma(x, y) = \kappa_\sigma(y, x)$ (symmetry). A frequently used kernel is the Gaussian kernel $\kappa_\sigma(x, y) = \frac{1}{\sqrt[d]{2\pi\sigma^d}} \exp(\frac{-\|x-y\|^2}{2\sigma^2})$, where σ is the bandwidth parameter of κ_σ.

If the data size $|Y| = n$, each estimation for point x using f_σ has $O(n)$ time complexity.

A. Adaptive KDE

To make KDE adaptive to local data distribution, there are two distinct approaches [1], where a fixed σ is replaced with either σ_x or σ_y. In the first approach, σ_x adapts to the test point x. In the second approach, σ_y adapts to every point in the given dataset.

Because the computational cost is high in finding the adapted bandwidth for each point, the first approach is often used as it needs to find the adapted bandwidth for the test point only [1]. For example, though using a Gaussian kernel, KDEOS [13] employs $\sigma_x = \min(mean_{y \in kNN(x)} dist(y, x), \epsilon)$, where the first term in the min function is the average distance of the k nearest neighbors of x found in the set Y, and the second term ϵ is a small positive constant to avoid zero bandwidth being assigned to σ_x. Here, the bandwidth parameter is replaced with k; yet a fixed setting of k allows the bandwidth to be adaptive to the local density of x.

[1]This is despite the fact that all adapted bandwidths can be found in the preprocessing step using the second approach.

KDEOS reduces the cost of each density estimation from $O(n)$ to $O(k)$ by using the k nearest neighbors only as follows:

$$f_{\sigma_x}(x|Y) = \frac{1}{k} \sum_{y \in kNN(x)} \kappa_{\sigma_x}(x, y)$$

However, the total complexity of KDEOS remains $O(n)$ because of the need to perform k-nearest-neighbour search.

There are other variable bandwidth KDEs (e.g., [1], [15], [16]), but none address the high computational cost issue.

B. Fast and non-adaptive KDE

There are a number of fast non-adaptive KDE methods, e.g., fastKDE [7] is based on an approximate Fourier transform; and it uses a non-uniform fast Fourier transform (FFT) to calculate the empirical characteristic function of a given dataset in a computationally efficient way. Many FFT methods rely on binning methods [4], [5] in order to reduce the number of accesses to kernel. Because of the use of bins, these methods are limited to few dimensions only as the number of bins grows exponentially with the number of dimensions [8].

An approach which is closer to ours is to use kernel functional approximation such as the Nyström method [9]. They aim to produce a finite-dimensional feature map that is an approximation to a feature map of intractable dimensionality of a kernel. With the finite-dimensional feature map, it can be used in KDE to reduce its runtime as in our approach, but at the expense of lower accuracy because of the approximation.

Yet another approach is to derive a small sample of points from a given dataset so that the estimation error as a result of using the subset is bounded by some constant. This approach includes coresets [17], kernel herding [18] and sparse approximation of kernel mean [19]. All these methods have a computationally expensive process to derive the sample before one can reap the benefit of a fast KDE. The first two methods cost $O(n^2)$. The sparse approximation method requires a matrix inversion which scales poorly with the sample size.

A more recent approach uses locality-sensitive hashing to provide sketches of the kernel summary (i.e., $\sum_{y \in Y} \kappa(x, y)$) in order to avoid the summation over all points in the dataset [10], [11], [20]. ACE [10] performs a kernel summary by using hash functions h_i as follows:

$$\sum_{i=1}^{t} A_i[h_i(x)|Y] = \sum_{y \in Y} \kappa(x, y)$$

where A_i is a sketch of Y, represented as an equi-width-bin histogram, associated with each h_i; and $h_i(x)$ hashes to a bin in the histogram, i.e., $A_i[h_i(x)] \subset Y$. The hash functions are from a family $\mathcal{F}$ such that $P_{h \sim \mathcal{F}}[h(x) = h(y)] = \kappa(x, y)$ [21]. The applicable kernels are p-stable LSH kernels [22] for Euclidean and Manhattan distances. Though the kernels have complex closed-form expressions, they are not explicitly used in deriving the hash functions. RACE [11] builds on top of ACE and uses the median of means as the final output.

An alternative method to build an LSH-based KDE is via sampling. A representative method is called Hashing based

Estimator (HBE) [23], [20], [12], which can employ commonly used kernels such as Laplacian and Exponential kernels. To estimate x, a uniformly random point y is selected from bin $B_i(x) = \{z \in Y | h_i(z) = h_i(x)\}$. The final estimation is given as follows:

$$f_B(x|Y) = \frac{1}{t} \sum_{i=1;\ y \sim B_i(x)}^{t} \frac{|B_i(x)|}{n} \sqrt{\kappa(x,y)}$$

The common ingredient of these LSH-based KDEs is that they employ a random projection [22], [11], to implement hash functions to map points in the data space to one-dimensional bins.

Though these LSH-based methods may be viewed as producing space partitions, they are a by-product of designing a hashing scheme which has collision probability equal to a similarity, often a kernel having a closed-form expression [22], [11]. The partitions created by LSH methods have no data dependent characteristic, unlike those used in IK mentioned above. They are fixed-width bins of a histogram in a space transformed by a random projection.

Section Summary and A Contrast with the Proposed Approach

One key point, which is often overlooked, is that the above approaches for both adaptive KDE and fast non-adaptive KDE assume that a *data independent kernel* is used, regardless of the methodologies employed.

In sharp contrast, we propose to use a *data dependent kernel* called Isolation Kernel (IK) to create an adaptive KDE. IK has a parameter, ψ, which corresponds to the bandwidth parameter in Gaussian kernel; yet, a fixed ψ allows IK to adapt to local data distribution to produce: **two points in a sparse region are more similar than two points of equal inter-point distance in a dense region** (see Lemma 1 in the next section.) This is in contrast to the translation invariant property of existing kernels such as Gaussian kernel and Laplacian kernel.

IK has two advantages compared with existing KDEs. First, existing KDEs use a data independent kernel which often has a feature map with intractable dimensionality. IK has a sparse and finite-dimensional feature map which is derived directly from data. This enables the resultant KDE to have $O(1)$ time complexity, without an expensive preprocessing to produce an approximate finite-dimensional feature map.

Second, using IK instead of κ_σ in Equation (1), a fast adaptive KDE is produced without additional (often computationally expensive) processes in finding the adaptive bandwidth or some representative summary of the dataset (in terms of either a subset of points or sketches or both.)

III. Data Dependent Isolation Kernel

Isolation Kernel [14] is first proposed as a data dependent kernel which adapts to the local density of the data distribution. It is defined as follows:

Let $D \subset \mathcal{X}_D \subset \mathbb{R}^d$ be a dataset sampled from an unknown distribution $\mathcal{P}_D$ where $\mathcal{X}_D$ is the support of $\mathcal{P}_D$ on $\mathbb{R}^d$. Let $\mathbb{H}_\psi(D)$ denote the set of all partitionings H that are admissible from $\mathcal{D} \subset D$, where each point has an equal probability of being selected; and $|\mathcal{D}| = \psi$. Each partition $\theta[z] \in H$ isolates a point $z \in \mathcal{D}$ from the rest of the points in $\mathcal{D}$. The union of all non-overlapping partitions of each partitioning covers the entire space.

Isolation Kernel [14], [24] is defined as follows.

Definition 1. *For any two points $x, y \in \mathbb{R}^d$, Isolation Kernel of x and y is defined to be the expectation taken over the probability distribution on all partitionings $H \in \mathbb{H}_\psi(D)$ that both x and y fall into the same isolating partition $\theta[z] \in H$, where $z \in \mathcal{D} \subset D$, $\psi = |\mathcal{D}|$:*

$$\kappa_\psi(x, y \mid D) = \mathbb{E}_{\mathbb{H}_\psi(D)}[\mathbb{1}(x, y \in \theta[z] \mid \theta[z] \in H)] \quad (2)$$

where $\mathbb{1}(\cdot)$ is an indicator function.

In practice, Isolation Kernel κ_ψ is constructed using a finite number of partitionings $H_i, i = 1, \ldots, t$, where each H_i is created using randomly subsampled $\mathcal{D}_i \subset D$; and θ is a shorthand for $\theta[z]$:

$$\begin{aligned} \kappa_\psi(x, y \mid D) &\simeq \frac{1}{t} \sum_{i=1}^{t} \mathbb{1}(x, y \in \theta \mid \theta \in H_i) \\ &= \frac{1}{t} \sum_{i=1}^{t} \sum_{\theta \in H_i} \mathbb{1}(x \in \theta) \mathbb{1}(y \in \theta) \end{aligned} \quad (3)$$

This gives a good approximation of $\kappa_\psi(x, y \mid D)$ when $|D|$ and t are sufficiently large to ensure that the ensemble is derived from a sufficient number of mutually independent $\mathcal{D}_i, i = 1, \ldots, t$.

Let $\rho_D(x)$ denote the density of $\mathcal{P}_D$ at point x. The unique characteristic of Isolation Kernel [14], [24] is given as follows:

Lemma 1. *$\forall x, y \in \mathcal{X}_\mathsf{S}$ and $\forall x', y' \in \mathcal{X}_\mathsf{T}$ such that $\forall_{z \in \mathcal{X}_\mathsf{S}, z' \in \mathcal{X}_\mathsf{T}}\ \rho_D(z) < \rho_D(z')$, Isolation Kernel κ_ψ has the unique characteristic that $\ell_p(x - y) = \ell_p(x' - y')$ implies*

$$\kappa_\psi(x, y \mid D) > \kappa_\psi(x', y' \mid D)$$

As a result, $\kappa_\psi(x, y \mid D)$ is not translation invariant.

A. Feature map of Isolation Kernel

Here we present the feature map in RKHS (Reproducing Kernel Hilbert Space) associated with Isolation Kernel κ_ψ based on partitionings $H_i, i = 1, \ldots, t$; and partitions $\theta_j \in H_i, j = 1, \ldots, \psi$.

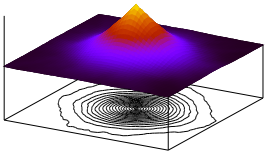
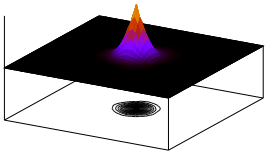
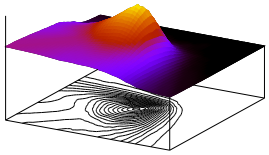
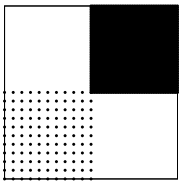
Given H_i, let $\Phi_i(x)$ be a ψ-dimensional binary column vector representing all $\theta_j \in H_i$, $j = 1, \ldots, \psi$; where x must fall into only one of the ψ partitions.

The j-component of the vector is: $\Phi_{ij}(x) = \mathbb{1}(x \in \theta_j \mid \theta_j \in H_i)$. Given t partitionings, $\Phi(x)$ is the concatenation of $\Phi_1(x), \ldots, \Phi_t(x)$.

Definition 2. ***Feature map of Isolation Kernel.*** *For point $x \in \mathbb{R}^d$, the feature mapping $\Phi : x \to \{0,1\}^{t \times \psi}$ of κ_ψ is a vector that represents the partitions in all the partitionings $H_i \in \mathbb{H}_\psi(D)$, $i = 1, \ldots, t$; where x falls into only one of the ψ partitions in each partitioning H_i.*

Let $\mathbb{1}$ be a shorthand of $\Phi_i(x)$ such that $\Phi_{ij}(x) = 1$ and $\Phi_{ik}(x) = 0, \forall k \neq j$ for any $j \in [1, \psi]$.

TABLE I: Kernel distributions of IK $\kappa_\psi(x, c)$, implemented using Voronoi diagram, for all $x \in \mathbb{R}^d$ of different ψ values, where $c = [0.5, 0.5]$ is a fixed point at the center of the data space. Data dependent IK derived from the dataset and an example Voronoi diagram are shown in the last two columns.

Data Independent IK: $\kappa_\psi(x, c\|U)$		Data Dependent IK: $\kappa_\psi(x, c\|D)$	Dataset	Voronoi Diagram
$\psi = 10$	$\psi = 100$	$\psi = 20$		
				

$\forall x \in \mathbb{R}^d$, the feature map $\Phi(x)$ is sparse because exactly t out of $t\psi$ elements are 1 and the rest are zero. This gives $\| \Phi(x) \| = \sqrt{t}$. Note that while every point has $\| \Phi(x) \| = \sqrt{t}$ and $\Phi_i(x) = \mathbb{1}$ for all $i \in [1, t]$, they are not all the same point in RKHS.

As a result, Isolation Kernel can be re-expressed as:

$$\kappa_\psi(x, y \mid D) \simeq \frac{1}{t} \langle \Phi(x|D), \Phi(y|D) \rangle .$$

B. Example kernel distributions

Examples of kernel distributions of $\kappa_\psi(x, y)$ derived from a dataset U of uniform density distribution (data independent IK) and a given dataset D (data dependent IK) are shown in Table I.

Note that ψ of IK corresponds to the bandwidth σ in Gaussian kernel, except that a large ψ produces sharp IK distribution; whereas a small σ yields a sharp Gaussian kernel distribution.

The key difference between the data independent IK and the data dependent IK is that the former is translation invariant and the latter is not. For the data dependent IK shown in Table I, a point in the left corner (the sparse region in the dataset used to derive the IK) is more similar to $c = [0.5, 0.5]$ than a point in the right corner (the dense region), when the two points have the same distance to c. This is a demonstration of Lemma 1.

Table II shows examples of kernel distributions of data dependent Isolation Kernel $\kappa_\psi(x, y|D)$ and the adaptive bandwidth Gaussian kernel $\kappa_{\sigma_c}(x, c)$, as used in KDEOS [13]. The latter adapts σ_c based on the average distance of the k nearest neighbours.

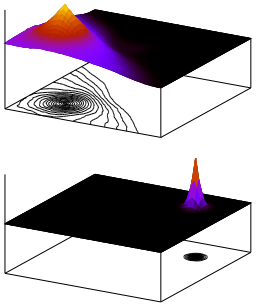
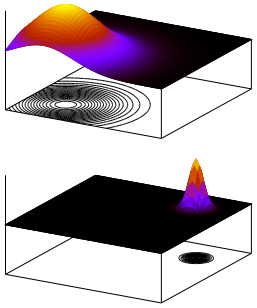
Isolation Kernel used in KDE has $O(1)$ time complexity for each estimation, after an one-off preprocessing of $O(n)$. In contrast, existing KDE that employs Gaussian kernel has $O(n)$ time complexity for each estimation.

In the next two sections, we present the proposed KDE and an analysis to show that it is a consistent estimator.

IV. Isolation Kernel Density Estimator

Given a dataset Y of points drawn from $\mathcal{P}_Y$, the kernel density estimator f_ψ which employs Isolation Kernel κ_ψ is given as follows:

TABLE II: Kernel distributions of data dependent Isolation Kernel $\kappa_\psi(x, c|D)$ using Voronoi diagram for all $x \in \mathbb{R}^d$; and adaptive bandwidth Gaussian kernel $\kappa_{\sigma_c}(x, c)$, as used in KDEOS. Two (fixed) reference points c are used: [0.24,0.24] for the first row and [0.76,0.76] for the second row. The same dataset shown in Table I is used here.

$\kappa_\psi(x, c\|D)$	$\kappa_{\sigma_c}(x, c)$
	

$$f_\psi(x|Y) = \frac{1}{|Y|} \sum_{y \in Y} \kappa_\psi(x, y|D) \simeq \frac{1}{t} \left\langle \Phi(x|D), \widehat{\Phi}(Y|D) \right\rangle \quad (4)$$

where $\widehat{\Phi}(Y|D) = \frac{1}{|Y|} \sum_{y \in Y} \Phi(y|D)$ is the *kernel mean map* of Y; and $\Phi(\cdot|D)$ is derived from D.

Note that $\widehat{\Phi}(Y|D)$ needs to be computed only once in a preprocessing. The dot product expression enables the time complexity to be reduced from $O(nm)$ to $O(n + m)$ for m point estimations.

We call this estimator `IKDE`. In practice, $Y = D$. However, it is possible that $Y \neq D$, i.e., IK is derived from a dataset, different from that which requires KDE estimations. This was demonstrated using a dataset of uniform density distribution to produce a data independent IK in the last section.

The data dependent Isolation Kernel $\kappa_\psi(x, y|D)$, which is derived from a dataset D, can be viewed as a kernel with local adaptive bandwidth. This is achieved with a single parameter ψ; unlike existing KDEs (with the exception of KDEOS [13].)

V. IKDE IS A CONSISTENT ESTIMATOR

We show in this section that IKDE is a consistent estimator under the following conditions.

Let U be a dataset generated from uniform density distribution $\mathcal{P}_U$ in $\mathcal{X}_U$ or the equivalent t sets of $\mathcal{U}$ of ψ points drawn from $\mathcal{P}_U$. The kernel is denoted as $\kappa_\psi(\cdot,\cdot|U)$; and it is implemented using multiple Voronoi diagrams [25] generated using $\mathcal{U} \subset U$, where each Voronoi partition θ is part of a Voronoi diagram H. This implementation is the same as used by [24], except that U is used to generate the Voronoi diagrams rather than the given dataset Y.

The generation of $\mathcal{U}$ consisting of uniformly distributed ψ points in $\mathcal{X}_U$ is approximated by a homogeneous Poisson point process of intensity $\psi/|\mathcal{X}_U|$, and the yielded Voronoi diagram is named a Poisson-Voronoi tessellation.

The expected volume of a Voronoi partition $\theta[z]$ generated by this process is known to be $|\mathcal{X}_U|/\psi$ [26], and thus

$$\begin{aligned}\int_{\mathbb{R}^d} \kappa_\psi(x,y \mid U)dy &= \int_{\mathbb{R}^d} \mathbb{E}_{\mathbb{H}_\psi(U)}[\mathbb{1}(x,y \in \theta[z] \mid \theta[z] \in H)]dy \\ &= \mathbb{E}_{\mathbb{H}_\psi(U)}\left[\int_{\mathbb{R}^d} \mathbb{1}(y \in \theta[z])dy \;\middle|\; x \in \theta[z] \in H\right] = \frac{|\mathcal{X}_U|}{\psi}.\end{aligned}$$

Based on this fact, we introduce the normalized version of $\kappa_\psi(x,y \mid U)$ to satisfy the normalization constraint.

$$\bar{\kappa}_\psi(x,y \mid U) = \frac{\psi}{|\mathcal{X}_U|}\kappa_\psi(x,y \mid U). \tag{5}$$

This also satisfies the symmetry, because the expected shape of $\theta[z]$ in the Poisson-Voronoi tessellation is isotropic according to the homogeneity of the point process [26].

By requiring that $\mathcal{X}_U$ covers $\mathcal{X}_Y$ and applying $\bar{\kappa}_\psi(x,y \mid U)$, Eq. (4) can be re-expressed as follows:

$$\begin{aligned}\bar{f}_\psi(x|Y) &= \frac{1}{|Y|}\sum_{y\in Y} \bar{\kappa}_\psi(x,y \mid U) \text{ s.t. } \mathcal{X}_Y \subseteq \mathcal{X}_U. \\ &\simeq \frac{\psi}{t|\mathcal{X}_U|}\left\langle \Phi(x|U), \widehat{\Phi}(Y|U)\right\rangle \end{aligned} \tag{6}$$

In the next subsection, we show that this estimator is a consistent estimator, so as the traditional KDE which employs Gaussian kernel, but its consistency is even stronger than the traditional KDEs.

A. Bias and variance of IKDE

Here, we show that IKDE using Voronoi diagrams is a consistent estimator, but the consistency is even stronger than the ordinary KDEs, *i.e.*, it is entailed by Lipschitz continuity of the target density function and some specific combination of $|Y|$ and ψ.

As mentioned earlier, the expected volume of any Voronoi partition $\theta[z] \in H$ generated by a homogeneous Poisson point process is the inverse of its intensity [26], *i.e.*,

$$\mathbb{E}_{\mathbb{H}_\psi(U)}[\ |\theta[z]|\ ] = |\mathcal{X}_U|/\psi. \tag{7}$$

This is simply because the probability that a point uniformly distributed in $\mathcal{X}_U$ falls into a partition is $1/\psi$. This reasoning immediately gives the following proposition.

Proposition 1. *For any pair of partitionings $H, H' \in \mathbb{H}_\psi(U)$, let two partitions $\theta[z] \in H$ and $\theta'[z'] \in H'$ have non-empty intersection $\theta[z] \cap \theta'[z'] \neq \emptyset$. The expected volume of the intersection is as follows:*

$$\mathbb{E}_{\mathbb{H}_\psi(U)}[\ |\theta[z] \cap \theta'[z']|\ ] = \frac{|\mathcal{X}_U|}{\psi^2}.$$

Proof. The probability that a point x uniformly distributed in $\mathcal{X}_U$ falls into $\theta[z]$ is $1/\psi$. This also holds for $\theta'[z']$. Because H and H' are independently generated in $\mathbb{H}_\psi(U)$, the probability that x belongs to both $\theta[z]$ and $\theta'[z']$, *i.e.*, $x \in \theta[z] \cap \theta'[z']$, is $1/\psi^2$. Accordingly, the proposition holds. □

Let $R_{max}(\psi)$ be the circumscribed radii of a Voronoi partition $\theta[z] \in H$ generated in a homogeneous Poisson point process of intensity $\psi/|\mathcal{X}_U|$. [27] showed that there exists a function $t(\psi)$ as $\forall\epsilon > 0, \lim_{\psi\to\infty} P(R_{max}(\psi) \leq \epsilon) = e^{-e^{-t(\psi)}}$, where $\lim_{\psi\to\infty} t(\psi) = \infty$.

This is rewritten as the following proposition.

Proposition 2. *$R_{max}(\psi)$ converges on 0 in probability as $\psi \to \infty$, i.e., $\forall\epsilon > 0, \lim_{\psi\to\infty} P(R_{max}(\psi) > \epsilon) = 0$.*

The data independent and normalized version of Isolation Kernel $\bar{\kappa}_\psi(x,y \mid U)$ defined by (2) and (5) is rewritten as follows.

$$\begin{aligned}\bar{\kappa}_\psi(x,y \mid U) &= \frac{\psi}{|\mathcal{X}_U|}\mathbb{E}_{\mathbb{H}_\psi(U)}[\mathbb{1}(x,y \in \theta[z] \mid \theta[z] \in H)] \\ &= \frac{\psi}{|\mathcal{X}_U|}\mathbb{E}_{\mathbb{H}_\psi(U)}[\mathbb{1}(y \in \theta[z] \mid x \in \theta[z] \in H)] \\ &= \frac{\psi}{|\mathcal{X}_U|}\sum_{H\in\mathbb{H}_\psi(U)} p(H)\int_{\theta[z|x]} \delta(r-y)dr,\end{aligned}$$

where $p(H)$ is the probability to draw H from $\mathbb{H}_\psi(U)$, $\theta[z|x]$ is $\theta[z] \in H$ containing x, and $\delta(\cdot)$ is the Dirac delta function.

Here, we introduce an assumption more widely applicable than the differentiability, i.e., ρ_Y is Lipschitz continuous in $\mathcal{X}_Y$. Based on this assumption, we derive the following theorem on bias and variance of IKDE.

Theorem 1. *Assuming Lipschitz continuity of $\rho_Y(y)$ for any $y \in \mathcal{X}_Y$, the bias and the variance of $\bar{f}_\psi(x|Y)$ based on Voronoi partitioning are bounded as follows, where L_Y is Lipschitz constant of $\rho_Y(y)$.*

$$\begin{aligned}Bias_\psi[\bar{f}_\psi(x|Y)] &\leq 2L_Y\mathbb{E}_{\mathbb{H}_\psi(U)}[R_{max}(\psi)], \\ Var_\psi[\bar{f}_\psi(x|Y)] &\leq \frac{\psi+2}{|Y||\mathcal{X}_U|}(\rho_Y(x) + 2L_Y\mathbb{E}_{\mathbb{H}_\psi(U)}[R_{max}(\psi)]).\end{aligned}$$

Both $Bias_\psi[\bar{f}_\psi(x|Y)]$ and $Var_\psi[\bar{f}_\psi(x|Y)]$ converge on 0 if $|Y| \to \infty$, $\psi \propto |Y|^\alpha$ and $0 < \alpha < 1$.

Thus, IKDE is a consistent estimator if $\psi \propto |Y|^\alpha$ and $0 < \alpha < 1$. The proof is given in Appendix B.

B. Relation to ordinary KDE

Given a true differentiable density function ρ_Y, [28] provides the bias and variance of KDE $f_\sigma(x|Y)$ which employs kernel $\kappa(\frac{x-y}{\sigma})$ as:

$$\begin{aligned}Bias_\sigma[f_\sigma(x|Y)] &\approx \alpha\sigma^2\nabla^2\rho_Y(x) \\ Var_\sigma[f_\sigma(x|Y)] &\approx \beta|Y|^{-1}\sigma^{-d}\rho_Y(x)\end{aligned}$$

where α and β are constants characterizing kernel κ.

These expressions imply that both bias and variance converge on 0 as $|Y| \to \infty$, *i.e.*, the KDE is consistent if $\sigma \propto |Y|^{-1/\eta}$ and $\eta > d$. The optimal estimator in terms of least mean integrated square error is one derived from $\eta = d + 4$ [28].

Similarly, Theorem 1 ensures the consistency of IKDE, if $|Y| \to \infty$, $\psi \propto |Y|^{\alpha}$ and $0 < \alpha < 1$. However, the major difference is that IKDE does not require differentiability of ρ_Y but Lipschitz continuity of ρ_Y which is a weaker condition than the former. This indicates the stronger consistency of IKDE than the ordinary KDE in terms of the applicability to density estimation.

C. Simulations of IKDE estimations

Our simulation in Figure 1 confirms that IKDE can estimate a Gaussian distribution well as in the case of the ordinary KDE (OKDE). It shows that IKDE is a consistent estimator: the estimate error decreases as the data size increases. We have also provided the estimations from two recent KDEs called RACE [11] and HBE [12]. They have approximately the same estimation error as IKDE.

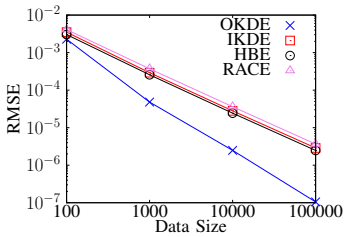


Fig. 1: Estimating Gaussian distribution $N(0, 1)$.

D. Different implementations of IK

IK can be implemented in different ways, as long as the isolation mechanism used produces large partitions in sparse regions and small partitions in dense regions [14], [24].

Here we provide one non-adaptive version and two adaptive versions of IKDE:

- $\texttt{IKDE}_U$ employs IK implemented using Voronoi Diagram, built from a uniformly distributed dataset (independent of the given dataset);
- $\texttt{IKDE}_V$ employs IK implemented using Voronoi Diagram, built from the given dataset [24];
- $\texttt{IKDE}_S$ employs IK implemented using hyperspheres, built from the given dataset [29].

$\texttt{IKDE}_U$ is used as a baseline to examine the utility of the adaptive versions $\texttt{IKDE}_V$ and $\texttt{IKDE}_S$.

VI. Time Complexities

To compute the densities of all points in a dataset of n points, a conventional KDE has $O(dn^2)$ time complexity, where d is the number of dimensions.

It is interesting to note that while the computational cost of each KDE estimation in KDEOS has been reduced from n data points to k nearest neighbours, the total time complexity is still $O(dn^2)$ because of the need to search for k nearest neighbours. The intervention step is as expensive as the original issue. RKDE [30] employs a kernelized least squared procedure to estimate the weight for each point used in KDE. Both this procedure and the final KDE computation have the same time complexity $O(dn^2)$.

A kernel functional approximation such as the Nyström method [31] produces an approximate finite-dimensional feature map of Gaussian kernel. It uses a subset of 'landmark' data points instead of all the data points. Given s landmark points, it has $O(ns^2)$. Then, the resultant KDE takes $O(ns)$ to compute the densities of all n points. Thus, the total time cost is $O(ns^2)$.

HBE [12] has processing time $O(nL \times min(d, d/\sigma))$ and query for n points costs $O(ndL)$, where L is the number of hash tables; and σ is the parameter for the Laplacian kernel. RACE [11] has time complexity as $O(n(d \log d + KL))$, where K is the number of bits used for a hash table.

All versions of IKDE have the same time complexity $O(dnt\psi)$, where t and ψ are constant parameters of IK. Note that it is possible that an appropriate setting of IK yields $t\psi > n$ on a particular dataset. In this case, IKDE runs slower than KDE with $O(dn^2)$. But in all large datasets we have used, $t\psi < n$. In addition, IKDE is amenable to GPU acceleration—reducing the cost to $O(dn\frac{t\psi}{\omega})$ with ω factor of parallelization. A GPU implementation provides a significant speedup, e.g., 930 CPU seconds and 33 GPU seconds on the Http dataset, having half a million data points.

In a nutshell, only IKDE, HBE and RACE have constant time for each estimation among existing KDEs.

VII. Using KDE for Anomaly Detection

Using KDE for anomaly detection is straightforward. Simply employ the density estimated for every point in a given dataset to rank these points. Anomalies are points that have the lowest densities in the dataset.

To compare with $\texttt{IKDE}_U$, $\texttt{IKDE}_V$ and $\texttt{IKDE}_S$, we use the following KDEs that employ a data independent kernel:

1) OKDE denotes the ordinary KDE (i.e., Eq 1);
2) OKDE_N which uses the Nyström method [31] to approximate the feature map;
3) Robust KDE called RKDE [30];
4) Adaptive KDE called KDEOS [13]; and
5) Two recent LSH-based KDEs: RACE [11] and HBE [12].

Furthermore, we also compare with (i) OCSVM [32] which employs the Gaussian kernel to examine whether an additional learning will help to improve the detection accuracy of KDE; (ii) iForest [33]; and (iii) LOF [34]. The detection accuracy is measured in terms of AUC (area under ROC curve). The experimental settings are stated in Appendix A.

A. Detection accuracy

The overall AUC result is shown in the second last row of Table III in terms of rank (i.e., the ranks of individual detectors averaged over all datasets.) $\texttt{IKDE}_S$, LOF and $\texttt{IKDE}_V$ are the three top ranked anomaly detectors, followed by HBE

TABLE III: Results of KDE based anomaly detectors, OCSVM, iForest and LOF (AUC). O/M is out of memory error. AE is arithmetic error. All methods use Gaussian kernel, except the three versions of IKDE, iForest and LOF; and RACE uses p-stable LSH kernels [22] for Euclidean distance & HBE uses Laplacian kernel (as used in their papers [11], [12].) The best AUCs are boldfaced in each dataset. The last row shows the runtimes of individual detectors on the largest dataset Http.

Dataset	#Points	#Ano	#Attr	iForest	OCSVM	LOF	OKDE	OKDE$_N$	RKDE	KDEOS	RACE	HBE	IKDE$_U$	IKDE$_V$	IKDE$_S$
Http	567497	2211	3	.99	.97	.5	.99	.98	O/M	.91	**1**	**1**	**1**	.99	**1**
ForestCover	286048	2747	10	.93	.96	**.98**	.85	.86	O/M	.54	.87	.91	.85	.93	.97
IoT_botnet	213814	1000	115	.94	.93	**1**	.83	.68	O/M	.69	.99	.97	.99	.96	**1**
Smtp	95156	30	3	.92	.91	**.96**	.81	.77	O/M	.81	.85	.86	.85	**.96**	**.96**
Muon	94066	500	50	.74	.75	.75	.63	.55	O/M	.53	.79	.82	.77	**.85**	.82
ALOI	50000	1508	27	.55	.53	.79	.60	.54	O/M	**.81**	.55	.58	.54	.71	.79
Shuttle	49097	3511	9	**.99**	.98	.58	.98	**.99**	O/M	.55	**.99**	**.99**	**.99**	.98	**.99**
Electron	37199	700	50	.80	.64	.77	.65	.57	O/M	**.81**	.58	.68	.56	.74	.77
Mammography	11183	260	6	.87	.84	.86	.85	.86	.72	.79	.86	**.88**	**.88**	.87	**.88**
MNIST_479	12139	50	784	.45	.59	.85	.69	.60	AE	**.86**	.54	.57	.48	.69	.82
MNIST_230	12117	10	784	.88	.96	**1**	.97	.97	AE	.97	.90	.94	.96	.97	.99
PenDigits	9868	20	17	.93	**.98**	.97	**.98**	.95	.86	.92	.89	.97	**.98**	**.98**	.97
EEG_eye	8422	165	14	.58	.54	.86	.55	.47	.62	**.88**	.55	.56	.59	.74	.86
Speech	3686	61	400	.46	.65	.73	.46	.47	.50	**.77**	.55	.59	.51	.50	.76
		Average rank		6.5	6.93	3.86	6.93	8.07	—	6.5	6.43	4.93	5.57	4.07	2.00
Http		Runtime (seconds)		24	15702	1319	9689	107	—	12346	586	2052	27	31	48

and IKDE$_U$. The three bottom-ranked detectors are OCSVM, OKDE and OKDE$_N$.

Some highlights are given below:

- IKDE$_S$ is the best KDE anomaly detector. It has better AUC than the best existing KDE, i.e., HBE, with 9 wins, 5 draws and no losses.
- IKDE$_S$ and LOF have comparable accuracy, where IKDE$_S$ has 5 wins, 6 draws and 3 losses.
- Existing 'improved' versions of KDE, including KDEOS, RACE and HBE, do not perform better substantially than the ordinary KDE. They have 7, 5 and 4 losses, respectively, against OKDE. As expected, OKDE$_N$ performs worse than OKDE on most datasets because it employs an approximate feature map of Gaussian kernel.
- Both IKDE$_V$ and IKDE$_S$ are better than iForest. They have at least 9 wins and a maximum of 2 losses.
- Among the two LSH-based KDEs, HBE has better accuracy but it runs slower than RACE.

B. Sensitivity study

Figure 2 (left) shows the boxplot comparing IKDE$_S$, RACE and HBE on the Muon dataset. This result shows that all detectors improve their AUC and decrease the variance of AUC as parameter t or L increases.

C. Parameter setting requirement: IKDE vs LOF

Here we provide an analysis of the parameter setting requirement of the two top-performing detectors, i.e., IKDE and LOF, which we have discovered in the last subsection.

Figure 2 (right) shows the best parameters used in IKDE and LOF, as the data size increases on the Http dataset. A huge difference emerges: LOF requires the k to be set proportional to the data size in order produce the best accuracy; whereas IKDE can use the same ψ parameter for all data sizes.

The above LOF behavior is the characteristic of kNN-based algorithms. In contrast, KDE-based algorithms such as RACE and HBE, like IKDE, have the best parameter setting which is usually not related to data size.

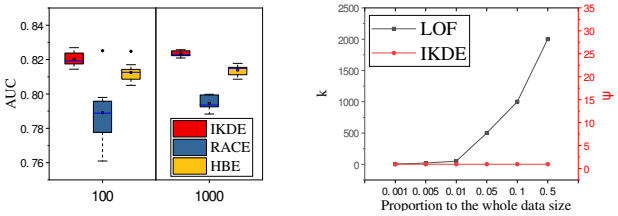


Fig. 2: Left: Boxplot of 10 runs for IKDE$_S$, RACE and HBE. Right: Best parameters of IKDE and LOF, as data size increases on the Http dataset.

D. Runtime and Scalep test

The last row of Table III shows the runtime of individual detectors on the largest dataset Http. Among the KDE-based detectors, the three versions of IKDE are the fastest, followed by the Nyström accelerated OKDE$_N$. Both OKDE and KDEOS have significantly slower runtime: two orders of magnitude slower than IKDE. The only existing detector which runs faster than IKDE is iForest; but they are in the same order.

Table IV shows a scaleup test result. IKDE runs orders of magnitude faster than KDEOS and LOF. The gap in runtime increases as the data size increases. Note that both IKDE$_V$ and IKDE$_S$ have approximately the same runtime.

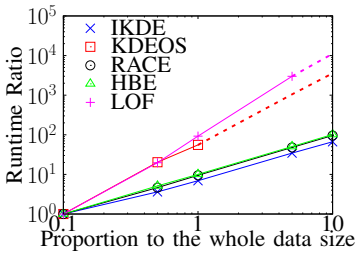
IKDE, RACE and HBE have virtually the same runtime ratio as shown in the graph in Table IV, i.e., they all have runtime linear to the data size.

Result Summary

First, IKDE is a successful adaptive multivariate KDE where the local adaptivity is achieved through the data dependent isolation mechanism of IK, controlled via one parameter ψ.

TABLE IV: Scaleup test: IKDE vs KDEOS, HBE and RACE on Http. Right graph: The y-axis is the runtime ratio with reference to the runtime on the 0.1 dataset size. The same parameter setting is used for each detector for all data sizes.

	Runtime (secs)	
Proportion	0.5	5
IKDE	25	237
LOF	282	42,687
RACE	287	2,968
HBE	1,049	10,342
KDEOS	4,505	233,000 projected



Second, $\texttt{IKDE}_S$ is the best detector which is fast and accurate. It has better accuracy on most datasets than all contenders, except LOF, and has the same time complexity as the fastest iForest.

Third, IKDE has two advantages over its closest contender LOF, i.e., LOF needs to increase k as data size increases, and has $O(n^2)$ time complexity. In contrast, IKDE usually does not need to change its best ψ setting as data size increases; and it has linear runtime. A consistent parameter setting, no matter how the data size increases, is very handy in practice, as it saves a huge amount of time for parameter tuning.

VIII. Discussion

On the surface, one may view IKDE as a kind of LSH-based KDE, when each isolating partition is regarded as an LSH 'bucket'. However, there are key differences. First, the shape of the 'buckets' depends on the isolation mechanism employed, which determines the kernel distribution. The reverse is true for LSH-based KDEs, the chosen (data independent) kernel determines the collision probability of the hash scheme. Second, no random projection is required for IK to produce these 'buckets'. Third, the number of 'buckets' ψ is IK's kernel parameter, whereas the number of 'buckets' L in LSH-based KDEs has no direct relation to the kernel. Fourth, the sizes of 'buckets' in IKDE are data dependent, but those in LSH are not. As shown in our results in Table III, the data dependency is the key in producing better anomaly detection accuracy.

As pointed out previously [24], the Voronoi diagram implementation of IK is amenable to GPU acceleration, so as the hypersphere implementation. This acceleration is required for the feature mapping only; and other processes in IKDE need no acceleration.

The first implementation of Isolation Kernel [14] employs iForest [33]. Though this implementation runs faster, IKDE that employs iForest provides lower detection accuracy than the nearest neighbor implementations we used here. In addition, it is not amenable to GPU implementation.

As indicated by [23], it may be possible to extend the data independent HBE to use data dependent LSH [35], [36]. This remains to be an open question. If this issue can be resolved, it would be interesting to examine the partitions of data dependent LSH whether they have the same characteristic of isolating partitions.

Some papers [37], [38] describe data dependent kernels in the context of semi-supervised learning where unlabeled data (in addition to label data) is used to modify the chosen data independent kernel. For example, a method [38] warps the distances between the Fourier features based on a data matrix from both labeled and unlabeled data. This method is not applicable to KDE where all data is unlabeled.

There is a strong relationship between KDE and kernel mean embedding (KME) [39]. The relationship between KDE and KME has also been stated by [17]. Applying IK in the context of KME has led to Isolation Distributional Kernel (IDK) [29]. It is of no surprise that IKDE anomaly detector is conceptually equivalent to IDK anomaly detector [29], though they are motivated from different sources: IDK is motivated from weaknesses in existing KME; and IKDE is motivated from weaknesses in existing KDEs.

IX. Conclusions

Our work resolves the age-old problem of producing an efficient adaptive KDE since the advent of KDE more than 60 years ago. Our use of Isolation Kernel (IK) is the key in providing three previously unseen advantages for kernel density estimation (KDE).

First, the data dependent IK with a fixed sharpness parameter achieves the same or better outcome of a data independent kernel with a bandwidth adjusted to local density. The adaptivity is an integral part of IK that needs no additional and expensive computation, unlike the use of a data independent kernel.

Second, the sparse finite-dimensional feature map, as a result of isolating partitions used to derive IK, contributes directly to $O(1)$ estimation time. Both advantages make IKDE the only adaptive KDE that has $O(1)$ time complexity.

Third, via a bias-variance analysis, we reveal that IKDE is a consistent estimator under less stringent assumptions than the ordinary KDE based on a Gaussian kernel.

In addition, in the context of anomaly detection, we identify two advantages over kNN-based detector LOF: (a) As data size increases, LOF must increase k setting, whereas IKDE runs on a constant ψ setting. (b) LOF has $O(n^2)$ time complexity, whereas IKDE potentially has sublinear complexity because its runtime is dominated by the feature mapping (which is linear to ψ).

We also demonstrate that two adaptive versions of IKDE are practical and constant-time density estimators. Both achieve the highest detection accuracy among all competing anomaly detectors; and they run orders of magnitude faster than existing Gaussian-kernel-based detectors. Only RACE, the fastest LSH-based KDE, has runtime comparable to IKDE, but RACE produces significantly weaker detection accuracy.

Acknowledgements

Kai Ming Ting is supported by Natural Science Foundation of China (62076120).

REFERENCES

[1] G. R. Terrell and D. W. Scott, "Variable kernel density estimation," *The Annals of Statistics*, vol. 20, no. 3, pp. 1236–1265, 1992.

[2] M. Rosenblatt, "Remarks on some nonparametric estimates of a density function," *The Annals of Mathematical Statistics*, vol. 27, no. 3, pp. 832–837, 1956.

[3] W. M. Gentleman and G. Sande, "Fast fourier transforms — for fun and profit," in *Proceedings of AFIPS fall joint computer conference*, 1966, pp. 563–578.

[4] D. W. Scott, "Using computer-binned data for density estimation," in *Proceedings of the 13th Symposium on the Interface of Computer Science and Statistics*. New York, NY: Springer US, 1981, pp. 292–294.

[5] B. W. Silverman, "Algorithm as 176: Kernel density estimation using the fast fourier transform," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 31, no. 1, pp. 93–99, 1982.

[6] M. P. Wand, "Fast computation of multivariate kernel estimators," *Journal of Computational and Graphical Statistics*, vol. 3, no. 4, pp. 433–445, 1994.

[7] T. A. O'Brien, K. Kashinath, N. R. Cavanaugh, W. D. Collins, and J. P. O'Brien, "A fast and objective multidimensional kernel density estimation method: fastkde," *Computational Statistics & Data Analysis*, vol. 101, pp. 148–160, 2016.

[8] D. W. Scott, *Multivariate Density Estimation: Theory, Practice, and Visualization*. 2nd Edition. Wiley, 2015.

[9] C. K. I. Williams and M. Seeger, "Using the Nyström method to speed up kernel machines," in *Advances in Neural Information Processing Systems 13*, T. K. Leen, T. G. Dietterich, and V. Tresp, Eds., 2001, pp. 682–688.

[10] C. Luo and A. Shrivastava, "Arrays of (locality-sensitive) count estimators (ACE): high-speed anomaly detection via cache lookups," in *Proceedings of The 2018 World Wide Web Conference*, 2017, pp. 1439–1448.

[11] B. Coleman and A. Shrivastava, "Sub-linear race sketches for approximate kernel density estimation on streaming data," in *Proceedings of The 2020 World Wide Web Conference*, 2020, pp. 1739–1749.

[12] A. Backurs, P. Indyk, and T. Wagner, "Space and time efficient kernel density estimation in high dimensions," in *Advances in Neural Information Processing Systems*, vol. 32, 2019, pp. 15 799–15 808.

[13] E. Schubert, A. Zimek, and H.-P. Kriegel, "Generalized outlier detection with flexible kernel density estimates," in *Proceedings of the SIAM Conference on Data Mining*, 2014, pp. 542–550.

[14] K. M. Ting, Y. Zhu, and Z.-H. Zhou, "Isolation kernel and its effect on SVM," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2018, pp. 2329–2337.

[15] S. R. Sain, "Multivariate locally adaptive density estimation," *Computational Statistics & Data Analysis*, no. 39, pp. 165–186, 2002.

[16] L. Zhang, J. Lin, and R. Karim, "Adaptive kernel density-based anomaly detection for nonlinear systems," *Knowledge-Based Systems*, vol. 139, pp. 50–63, 2018.

[17] J. M. Phillips and W. M. Tai, "Near-optimal coresets of kernel density estimates," *Discrete & Computational Geometry*, vol. 63, pp. 867–887, 2020.

[18] Y. Chen, M. Welling, and A. Smola, "Super-samples from kernel herding," in *Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence*, 2010, pp. 109–116.

[19] E. C. Cortes and C. Scott, "Sparse approximation of a kernel mean," *IEEE Transactions on Signal Processing*, vol. 65, no. 5, pp. 1310–1323, 2017.

[20] P. Siminelakis, K. Rong, P. Bailis, M. Charikar, and P. Levis, "Rehashing kernel evaluation in high dimensions," in *Proceedings of the 36th International Conference on Machine Learning*, 2019, pp. 5789–5798.

[21] M. S. Charikar, "Similarity estimation techniques from rounding algorithms," in *Proceedings of the 34th ACM Symposium on Theory of Computing*, 2002, pp. 380–388.

[22] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, "Locality-sensitive hashing scheme based on p-stable distributions," in *Proceedings of the 20th Annual Symposium on Computational Geometry*, 2004, pp. 253–262.

[23] M. Charikar and P. Siminelakis, "Hashing-based-estimators for kernel density in high dimensions," in *2017 IEEE 58th Annual Symposium on Foundations of Computer Science*, 2017, pp. 1032–1043.

[24] X. Qin, K. M. Ting, Y. Zhu, and V. C. S. Lee, "Nearest-neighbour-induced isolation similarity and its impact on density-based clustering," in *Proceedings of The Thirty-Third AAAI Conference on Artificial Intelligence*, 2019, pp. 4755–4762.

[25] F. Aurenhammer, "Voronoi diagrams—a survey of a fundamental geometric data structure," *ACM Computing Surveys*, vol. 23, no. 3, pp. 345–405, 1991.

[26] K. Alishahi and M. Sharifitabar, "Volume degeneracy of the typical cell and the chord length distribution for poisson-voronoi tessellations in high dimensions," *Advances in Applied Probability*, vol. 40, no. 4, pp. 919–938, 2008.

[27] P. Calka and N. Chenavier, "Extreme values for characteristic radii of a poisson-voronoi tessellation," *Extremes*, vol. 17, no. 3, pp. 359–385, 2014.

[28] B. W. Silverman, *Density estimation for statistices and data analysis*. Chapman and Hall, 1986.

[29] K. M. Ting, B.-C. Xu, T. Washio, and Z.-H. Zhou, "Isolation distributional kernel: A new tool for kernel based anomaly detection," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2020, pp. 198–206.

[30] J. Kim and C. D. Scott, "Robust kernel density estimation," *Journal of Machine Learning Research*, vol. 13, no. 82, pp. 2529–2565, 2012.

[31] C. Musco and C. Musco, "Recursive sampling for the nystrom method," in *Advances in Neural Information Processing Systems*, 2017, pp. 3833–3845.

[32] B. Schölkopf, J. C. Platt, J. C. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Computing*, vol. 13, no. 7, pp. 1443–1471, 2001.

[33] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *Proceedings of the IEEE International Conference on Data Mining*, 2008, pp. 413–422.

[34] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying density-based local outliers," in *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, 2000, pp. 93–104.

[35] A. Andoni and I. Razenshteyn, "Optimal data-dependent hashing for approximate near neighbors," in *Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing*, 2015, pp. 793–801.

[36] A. Andoni, T. Laarhoven, I. Razenshteyn, and E. Waingarten, "Optimal hashing-based time-space trade-offs for approximate near neighbors," in *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, 2017, pp. 47–66.

[37] G. Lever, T. Diethe, and J. Shawe-Taylor, "Data dependent kernels in nearly-linear time," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, 2012, pp. 685–693.

[38] C. Ionescu, A. Popa, and C. Sminchisescu, "Large-scale data-dependent kernel approximation," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, 2017, pp. 19–27.

[39] K. Muandet, K. Fukumizu, B. Sriperumbudur, and B. Schölkopf, "Kernel mean embedding of distributions: A review and beyond," *Foundations and Trends in Machine Learning*, vol. 10 (1–2), pp. 1–141, 2017.

APPENDIX

A. Experimental settings

The experiments ran on a Linux CPU machine: AMD 16-core CPU with each core running at 2 GHz and 32 GB RAM. The IK feature mapping ran on a machine having GPU: 2 x GTX 1080 Ti with 3584 (1.6 GHz) CUDA cores & 12GB graphic memory; and CPU: i9-7900X 3.30GHz processor (20 cores), 64GB RAM. Only the feature mapping of IK was conducted in GPU. Other processes in `IKDE` were run in CPU.

We use the code of Isolation Kernel available at https://github.com/IsolationKernel/Codes/tree/main/IDK/IKDE to build `IKDE`; and the code of LOF at https://github.com/scikit-learn/scikit-learn/blob/15a949460/sklearn/neighbors/_lof.py. Other codes are from the respective authors.

The reported results for randomized algorithms, `IKDE`, RACE and HBE are averaged over five trials. The results of the deterministic algorithms e.g., OKDE and KDEOS are obtained from a single trial.

All datasets are normalized to [0, 1] in the preprocessing.

Sources of datasets used are:

* lapad-web.icmc.usp.br/repositories/outlier-evaluation/DAMI,
* odds.cs.stonybrook.edu, sites.google.com/site/gspangsite,
* https://www.csie.ntu.edu.tw/ cjlin/libsvmtools/datasets/

The parameter search ranges are shown in Table V.

TABLE V: Parameter search ranges.

Algorithm	Parameter search ranges
IKDE, iForest	$\psi \in \{2^q \mid q = 1, \ldots, 12\}$; $t = 100$
OKDE$_N$	$\sigma \in \{2^m \mid m = -5, \ldots, 5\}$; $s = \sqrt{n}$
OKDE,RKDE,OCSVM	$\sigma \in \{2^m \mid m = -5, \ldots, 5\}$
KDEOS	$\sigma \in \{2^m \mid m = -5, \ldots, 5\}$; plus k as in LOF
LOF	$k \in \{1, 3, 5, 7, 11, 21, 51, 101, 201, 501, 1001, 2001\}$
RACE	$\epsilon = 0.05$; $R = \{2^g \mid g = 3, \ldots, 10\}$; plus HBE settings
HBE	$L = 100$; $\sigma = \{2^m \mid m = -5, \ldots, 5\}$

B. Proof of Theorem 1

Proof. Since all $y \in Y$ is $y \sim \mathcal{P}_Y$, we derive the following expectations.

$$\begin{aligned}
&\mathbb{E}_{\rho_Y}[\bar{\kappa}_\psi(x, y \mid U)] \\
&= \int_{\mathcal{X}_Y} \rho_Y(y) \frac{\psi}{|\mathcal{X}_U|} \sum_{H \in \mathbb{H}_\psi(U)} p(H) \int_{\theta[z|x]} \delta(r - y) dr dy \\
&= \frac{\psi}{|\mathcal{X}_U|} \sum_{H \in \mathbb{H}_\psi(U)} p(H) \int_{\mathcal{X}_Y} \rho_Y(y) \int_{\theta[z|x]} \delta(r - y) dr dy \\
&= \frac{\psi}{|\mathcal{X}_U|} \sum_{H \in \mathbb{H}_\psi(U)} p(H) \int_{\theta[z|x]} \rho_Y(y) dy
\end{aligned}$$

$$\begin{aligned}
&\mathbb{E}_{\rho_Y}[\bar{\kappa}_\psi(x, y \mid U)^2] \\
&= \int_{\mathcal{X}_Y} \rho_Y(y) \left\{ \frac{\psi}{|\mathcal{X}_U|} \sum_{H \in \mathbb{H}_\psi(U)} p(H) \int_{\theta[z|x]} \delta(r - y) dr \right\}^2 dy \\
&= \frac{\psi^2}{|\mathcal{X}_U|^2} \sum_{H \in \mathbb{H}_\psi(U)} p(H) \int_{\mathcal{X}_Y} \rho_Y(y) \left(\int_{\theta[z|x]} \delta(r - y) dr \right)^2 dy \\
&\quad + \frac{2\psi^2}{|\mathcal{X}_U|^2} \sum_{H, H' \in \mathbb{H}_\psi(U)} p(H) p(H') \\
&\quad \times \int_{\mathcal{X}_Y} \rho_Y(y) \int_{\theta[z|x]} \delta(r - y) dr \int_{\theta'[z|x]} \delta(r' - y) dr' dy \\
&= \frac{\psi^2}{|\mathcal{X}_U|^2} \sum_{H \in \mathbb{H}_\psi(U)} p(H) \int_{\mathcal{X}_Y} \rho_Y(y) \int_{\theta[z|x]} \delta(r - y) dr dy \\
&\quad + \frac{2\psi^2}{|\mathcal{X}_U|^2} \sum_{H, H' \in \mathbb{H}_\psi(U)} p(H) p(H') \int_{\mathcal{X}_Y} \rho_Y(y) \int_{\theta[z|x] \cap \theta'[z'|x]} \delta(r - y) dr dy \\
&= \frac{\psi^2}{|\mathcal{X}_U|^2} \sum_{H \in \mathbb{H}_\psi(U)} p(H) \int_{\theta[z|x]} \rho_Y(y) dy \\
&\quad + \frac{2\psi^2}{|\mathcal{X}_U|^2} \sum_{H, H' \in \mathbb{H}_\psi(U)} p(H) p(H') \int_{\theta[z|x] \cap \theta'[z'|x]} \rho_Y(y) dy
\end{aligned}$$

By Lipschitz continuity of $\rho_Y(y)$ and $x, y \in \theta[z]$, the following relation holds in $\theta[z]$.

$$\rho_Y(y) - \rho_Y(x) \leq |\rho_Y(y) - \rho_Y(x)| \leq L_Y |y - x| \leq 2L_Y R_{max}(\psi). \quad (8)$$

These expressions give the following bias and variance of $\bar{f}_\psi(x|Y)$.

$$\begin{aligned}
&Bias_\psi[\bar{f}_\psi(x|Y)] = \mathbb{E}_{\rho_Y}[\bar{\kappa}_\psi(x, y \mid U)] - \rho_Y(x) \\
&= \mathbb{E}_{\rho_Y}[\bar{\kappa}_\psi(x, y \mid U)] - \frac{\psi}{|\mathcal{X}_U|} \mathbb{E}_{\mathbb{H}_\psi(U)}[\, |\theta[z|x]| \,] \rho_Y(x) \\
&= \frac{\psi}{|\mathcal{X}_U|} \sum_{H \in \mathbb{H}_\psi(U)} p(H) \int_{\theta[z|x]} (\rho_Y(y) - \rho_Y(x)) dy \\
&\leq \frac{\psi}{|\mathcal{X}_U|} \sum_{H \in \mathbb{H}_\psi(U)} p(H) \int_{\theta[z|x]} 2L_Y R_{max}(\psi) dr \\
&= \frac{2L_Y \psi}{|\mathcal{X}_U|} \sum_{H \in \mathbb{H}_\psi(U)} p(H) R_{max}(\psi) \, |\theta[z|x]| \\
&\leq \frac{2L_Y \psi}{|\mathcal{X}_U|} \mathbb{E}_{\mathbb{H}_\psi(U)}[R_{max}(\psi)] \mathbb{E}_{\mathbb{H}_\psi(U)}[|\theta[z|x]|] \\
&= 2L_Y \mathbb{E}_{\mathbb{H}_\psi(U)}[R_{max}(\psi)].
\end{aligned}$$

The second line is from (7) and the homogeneity of $\theta[z]$, the forth line is from (8), and the last two lines are from Cauchy-Schwarz inequality and (7).

$$\begin{aligned}
&Var_\psi[\bar{f}_\psi(x|Y)] \\
&= \frac{1}{|Y|} (\mathbb{E}_{\rho_Y}[\bar{\kappa}_\psi(x, y \mid U)^2] - \mathbb{E}_{\rho_Y}[\bar{\kappa}_\psi(x, y \mid U)]^2) \\
&\leq \frac{1}{|Y|} \mathbb{E}_{\rho_Y}[\bar{\kappa}_\psi(x, y \mid U)^2] \\
&\leq \frac{\psi^2}{|Y||\mathcal{X}_U|^2} \sum_{H \in \mathbb{H}_\psi(U)} p(H) \int_{\theta[z|x]} (\rho_Y(x) + 2L_Y R_{max}(\psi)) dy \\
&\quad + \frac{2\psi^2}{|Y||\mathcal{X}_U|^2} \sum_{H, H' \in \mathbb{H}_\psi(U)} p(H) p(H') \\
&\quad \times \int_{\theta[z|x] \cap \theta'[z'|x]} (\rho_Y(x) + 2L_Y R_{max}(\psi)) dy \\
&= \frac{\rho_Y(x) \psi^2}{|Y||\mathcal{X}_U|^2} \mathbb{E}_{\mathbb{H}_\psi(U)}[|\theta[z|x]|] \\
&\quad + \frac{2\rho_Y(x) \psi^2}{|Y||\mathcal{X}_U|^2} \mathbb{E}_{\mathbb{H}_\psi(U) \times \mathbb{H}_\psi(U)}[\left|\theta[z|x] \cap \theta'[z'|x]\right|] \\
&\quad + \frac{2L_Y \psi^2}{|Y||\mathcal{X}_U|^2} \mathbb{E}_{\mathbb{H}_\psi(U)}[R_{max}(\psi) \, |\theta[z|x]|] \\
&\quad + \frac{4L_Y \psi^2}{|Y||\mathcal{X}_U|^2} \mathbb{E}_{\mathbb{H}_\psi(U) \times \mathbb{H}_\psi(U)}[R_{max}(\psi) \left|\theta[z|x] \cap \theta'[z'|x]\right|] \\
&\leq \frac{\rho_Y(x) \psi^2}{|Y||\mathcal{X}_U|^2} \mathbb{E}_{\mathbb{H}_\psi(U)}[|\theta[z|x]|] \\
&\quad + \frac{2\rho_Y(x) \psi^2}{|Y||\mathcal{X}_U|^2} \mathbb{E}_{\mathbb{H}_\psi(U) \times \mathbb{H}_\psi(U)}[\left|\theta[z|x] \cap \theta'[z'|x]\right|] \\
&\quad + \frac{2L_Y \psi^2}{|Y||\mathcal{X}_U|^2} \mathbb{E}_{\mathbb{H}_\psi(U)}[R_{max}(\psi)] \mathbb{E}_{\mathbb{H}_\psi(U)}[|\theta[z|x]|] \\
&\quad + \frac{4L_Y \psi^2}{|Y||\mathcal{X}_U|^2} \mathbb{E}_{\mathbb{H}_\psi(U)}[R_{max}(\psi)] \mathbb{E}_{\mathbb{H}_\psi(U) \times \mathbb{H}_\psi(U)}[\left|\theta[z|x] \cap \theta'[z'|x]\right|] \\
&= \frac{\psi + 2}{|Y||\mathcal{X}_U|} (\rho_Y(x) + 2L_Y \mathbb{E}_{\mathbb{H}_\psi(U)}[R_{max}(\psi)]).
\end{aligned}$$

The inequality of the forth line is from (8). The last two lines are from Cauchy-Schwarz inequality, (7) and Proposition 1.

By proposition 2, $Bias_\psi[\bar{f}_\psi(x|Y)] \to 0$ if $\psi \to \infty$. Accordingly, both $Bias_\psi[\bar{f}_\psi(x|Y)]$ and $Var_\psi[\bar{f}_\psi(x|Y)]$ converge on 0 if $|Y| \to \infty$, $\psi \propto |Y|^\alpha$ and $0 < \alpha < 1$. □