A Framework for Exploiting Local Information to Enhance Density Estimation of Data Streams

ARNOLD P. BOEDIHARDJO, U.S. Army Corps of Engineers
CHANG-TIEN LU, Virginia Tech
BINGSHENG WANG, Virginia Tech

The Probability Density Function (PDF) is the fundamental data model for a variety of stream mining algorithms. Existing works apply the standard nonparametric Kernel Density Estimator (KDE) to approximate the PDF of data streams. As a result, the stream-based KDEs cannot accurately capture complex local density features. In this article, we propose the use of Local Region (LRs) to model local density information in univariate data streams. In-depth theoretical analyses are presented to justify the effectiveness of the LR-based KDE. Based on the analyses, we develop the General Local rEgion AlgorithM (GLEAM) to enhance the estimation quality of structurally complex univariate distributions for existing stream-based KDEs. A set of algorithmic optimizations is designed to improve the query throughput of GLEAM and to achieve its linear order computation. Additionally, a comprehensive suite of experiments was conducted to test the effectiveness and efficiency of GLEAM.

Categories and Subject Descriptors: H.2.8 [**Database Applications**]: Data Mining

General Terms: Data Stream, Data Mining, Density Estimation

Additional Key Words and Phrases: Local region information, General Local rEgion AlgorithM (GLEAM)

ACM Reference Format:
Arnold P. Boedihardjo, Chang-Tien Lu, and Bingsheng Wang. 2014. A framework for exploiting local information to enhance density estimation of data streams. ACM Trans. Knowl. Discov. Data. 9, 1, Article 2 (August 2014), 38 pages.
DOI: http://dx.doi.org/10.1145/2629618

1. INTRODUCTION

Data stream mining has become a popular focus in data mining research. One of the primary reasons for the popularity of data streams is their pervasiveness in a variety of domains. Data streams can be categorized into two classes: multivariate and univariate streams. A multivariate stream is a multidimensional tuple ordered on, for example, the temporal dimension. An example of a multivariate stream is a spatiotemporal trajectory dataset composed of a moving object's latitude/longitude coordinate locations indexed by time. A univariate stream is defined as a one-dimensional tuple set. Univariate streams can be observed in a variety of domains: for example, in finance, data streams exist as the continually changing prices of a traded stock; in transportation, they exist as the detected vehicle volume measures within a roadway segment; and in medicine, they exist as the observed contraction rate of a human heart. These examples

Authors' addresses: A. P. Boedihardjo, Geospatial Research Lab – Engineer Research and Development Center, U.S. Army Corps of Engineers, 7701 Telegraph Rd, Alexandria, VA 22315; C.-T. Lu, Department of Computer Science, Virginia Tech, 7054 Haycock Road, Falls Church, VA 22043; Bingsheng Wang, Department of Computer Science, Virginia Tech, 7054 Haycock Road, Falls Church, VA 22043.

DOI: http://dx.doi.org/10.1145/2629618

demonstrate the breadth of scenarios for which univariate data stream applications can exist. Many more concrete examples of univariate data streams can be seen from the time series domain [Keogh et al. 2008] whereby one can apply important mining tasks such as outlier detection [Subramaniam et al. 2006] and pattern discovery [Wegman and Marchette 2003] to elicit additional knowledge from the data-generating process. In the case of multivariate data streams, one can employ dimension reduction to generate projected data streams for which univariate analytical techniques can be applied [Silverman 1986]. Univariate analysis is also useful for uncovering patterns in domains beyond the traditional Euclidean space. For example, there has been increasing interest in applying univariate probability density estimation on network data [Okabe et al. 2009; Xie and Yan 2008]. In fact, these recent works demonstrate that a univariate KDE in network space can produce enhanced estimates over a Euclidean multidimensional KDE. Hence, due to the wide-ranging applicability of univariate stream analysis, this work investigates and develops univariate analytical methods for data streams.

For many stream mining techniques, especially those based on a statistical framework, the Probability Density Function (PDF) is the principal employed data model [Aggarwal and Yu 2007; Babcock et al. 2002; Heinz and Seeger 2008; Wegman and Marchette 2003]. The PDF gives a complete distributional description of a random process and thus provides the basis for several mining algorithms. For instance, the PDF has been utilized as the core data model to support stream-based outlier detection, concept drift analysis, and pattern discovery. In outlier detection, the PDF is used to model a sensor's data distribution and estimate a distance-based outlier score of incoming sample points [Knorr and Ng 1998; Subramaniam et al. 2006]. For concept drift analysis, the PDF is utilized to develop, describe, and compare behavioral profiles of incoming data [Aggarwal 2003]. Last, in pattern discovery, the PDF is applied to visualize and uncover predictive structures in Internet packet data [Wegman and Marchette 2003].

In the data stream environment, the form of the stream's PDF (e.g., Gaussian, Poisson) is generally unknown and evolving. Under this context, a nonparametric estimation approach can be employed to estimate the PDF. A well established and effective nonparametric technique is the Kernel Density Estimator (KDE) [Hardle et al. 2004]. The formulation of the univariate KDE is as follows: for n independent and identically distributed (i.i.d) sample points $z_1, \ldots, z_n$, bandwidth h, and a kernel function $K(\cdot)$, the standard KDE is

$$\hat{f}_{KDE}(x) = \frac{1}{n}\sum_{i=1}^{n} K_h(x - z_i). \tag{1}$$

where $K_h(x - z_i) = \frac{1}{h}K(\frac{x-z_i}{h})$ and $\int K(t)dt = 1$.

From Equation (1), a KDE generates a probability estimate by summing the contributions at point x of the kernel functions $K(\cdot)$ that are superimposed and centered on samples $z_1, \ldots, z_n$. The bandwidth h controls the width of the kernels, and, as a result, the selection of the bandwidth h and kernel function $K(\cdot)$ significantly impacts the quality of estimate. Between these two parameters, it has been analytically and empirically shown that the estimation quality of the KDE is critically dependent on the bandwidth, whereas different kernel functions may provide marginal effects on the overall estimation quality [Hardle et al. 2004; Scott 1992; Silverman 1986]. In the standard KDE, a single bandwidth is applied globally; that is, the same bandwidth h is applied to each kernel $K(\cdot)$ centered on samples $z_1, \ldots, z_n$. This global/single bandwidth assignment can be problematic for a variety of distributions due to its inability to accurately and precisely model the local structures/features (e.g., modes of the distribution). Therefore, for complex densities such as the multimodal distribution of highway traffic speed during a multi-incident event, the global bandwidth KDE may produce poor and unsatisfactory estimation results [Sain 1994; Van Kerm 2003]. Estimation of local structures could be improved if the bandwidth h is allowed to vary across different regions of the density (e.g., smaller bandwidth for highly oscillatory section of the density).

Due to computational considerations, the majority of stream-based KDEs applies the global bandwidth form. As a consequence, these methods tend to generate inaccurate estimates of local structures within complex distributions because their bandwidth cannot be independently adjusted to model different regions of the density. The emphasis of existing stream-based KDEs has been on developing techniques for constructing and maintaining a finite-size model of the data stream without considering the bandwidth and its impact to estimation quality. Specifically, given a data stream of size n, the objective of existing techniques is to reduce the data stream to M representative objects where $M \ll n$. The standard global bandwidth KDE is then applied to the summarized objects to generate the PDF estimates. This reduced representation allows the KDE to be maintained in a fixed memory environment and gives rise to various $O(M)$ density query algorithms. However, because these techniques apply the standard KDE, they exhibit the same estimation problem inherent to all global bandwidth approaches as described earlier.

This article aims to address the estimation issue associated with the globally assigned bandwidth. In particular, accurate estimation of the local features cannot be achieved with a global/single bandwidth approach [Sain 1994; Van Kerm 2003]. To address this issue, we propose the use of Local Regions (LRs) to efficiently and effectively model the local structures within the PDF in $O(M)$ time. LRs are partitions of the sample set that associate a unique bandwidth to each partition. Based on LRs, a generalized adaptive bandwidth assignment algorithm is proposed. The proposed algorithm can be applied to an arbitrary class of global bandwidth stream-based KDEs to enhance its estimation accuracy and assure a worst-case density query cost of $O(M)$. Specifically, this article makes the following contributions:

1. *Theoretical analyses of the LR bandwidth approach:* Rigorous analyses of the LR-based KDE's expected error, asymptotic consistency, and convergence rate are presented. The results of the analyses are used to compare the LR approach to the standard KDE.
2. *Design of a generalized LR algorithm:* The General Local rEgion AlgorithM (GLEAM) is proposed for application to existing stream-based KDEs to enhance the estimation accuracy of complex distributions. Analyses of GLEAM's time and space complexity are also given.
3. *Development of optimization techniques:* Two optimization techniques (*heap-based regularization* and *hybrid kernel aggregation and filtering*) are proposed to improve GLEAM's query processing throughput. In addition, cost analyses of the proposed optimizations are provided.
4. *Comprehensive experiments to validate the effectiveness and efficiency:* GLEAM was applied to a set of representative stream-based KDEs, and the results showed that the LR algorithm improved the estimation quality of existing KDEs. Furthermore, GLEAM's throughput performances were comparable or better than its non-LR counterparts.

This article is organized as follows: Section 2 reviews fundamental properties of the KDE. Section 3 surveys related works of stream-based KDEs. Section 4 provides in-depth theoretical analyses of the LR-based KDE. We describe our proposed GLEAM approach and its associated optimizations in Section 5. Section 6 gives the experimental results and discussion. Final conclusions are drawn in Section 7.

2. BACKGROUND

The following provides a background on the KDE. The standard KDE (Equation (1)) at the query point x is essentially a summation of the contributions of weighting functions $K(\cdot)$ centered on each of the sample points. The constraint $\int K(t)dt = 1$ enforces the KDE to be a PDF because it results in $\int \hat{f}_{KDE}(x)dx = 1$. Most of the kernel functions used in practice are symmetric (e.g., Gaussian kernel), hence the KDE can be intuitively viewed as the summation of contributing portions of symmetric "bumps" that are centered above each data sample. The bandwidth h is regarded as the width of the kernel functions or "bumps." Therefore, h serves as the smoothing parameter of the resulting KDE: A large h can generate a smooth shape density, whereas a small h can provide an undersmoothed estimate.

The drawback of the standard KDE is its necessary inclusion for the use of a global bandwidth. Due to the existence of local features in PDFs, a global bandwidth may not be sufficient to model complex density structures (e.g., multimodal distributions). Examples of local features are differently shaped and sized modes within a PDF that arise in highway traffic data and financial trade stocks. However, the KDE still possesses several attractive features that include rigorous mathematical foundation; generalization to other density estimators, such as orthogonal series and histograms; asymptotic consistency; and inheritance of the kernel function's continuity and differentiability properties [Scott 1992; Silverman 1986]. Therefore, if the estimation issue associated with the global bandwidth can be resolved, the impact of the KDE on stream mining would be significantly enhanced.

The majority of stream-based KDEs employs the global bandwidth form (i.e., standard KDE), and, in particular, they utilize a specific bandwidth function called the Scott's Rule [Aggarwal 2003; Heinz and Seeger 2006; Silverman 1986; Subramaniam et al. 2006]. The Scott's Rule bandwidth assumes a Gaussian distribution reference and thus has a tendency to mask local features and oversmooth complex densities. This choice of bandwidth can have negative consequences to data stream applications. Suppose that the Scott's Rule bandwidth-based KDE is used for the task of detecting distance-based outliers [Subramaniam et al. 2006]; then, the generated distributional model can be oversmoothed and lead some estimates within the sparse regions to be biased upward [Boedihardjo et al. 2008]. This effect can cause the detection scheme to dismiss some true outliers. For mission-critical applications such as military surveillance, a single false dismissal can be disastrous.

In the following paragraphs, we briefly sketch the central issue regarding the Scott's Rule. The Scott's Rule bandwidth form is provided as follows:

$$h_{SR} = C\sigma_D \sqrt[-5]{|D|}, \tag{2}$$

where C is a constant that depends on the employed kernel function $K(\cdot)$ (e.g., $C \approx 1.06$ for Gaussian kernel and $C = \sqrt{5}$ for Epanechnikov kernel), D is an i.i.d data sample set, and σ_D is the standard deviation of D.

Due to the dependency on a single statistic σ_D to describe the *complete span* of the density, the Scott's Rule can fail to accurately estimate highly complex structures. Consider the case when D is a binormal density and each of its modes has a constant spread; then, σ_D increases with the distance of the two modes. As a result, σ_D inflates the Scott's Rule bandwidth value and causes the KDE to oversmooth. However, if the density is a simple unimodal structure, then σ_D can accurately describe the density structure, and the Scott's Rule will provide good estimates. The problem described above highlights the essential issue underlying all global bandwidth KDE techniques i.e., its inability to appropriately represent the complete density structure using a single measure.

Because of the drawbacks associated with the globally defined bandwidth, the Adaptive KDE (AKDE) was proposed to support locally varying bandwidths. The AKDE is defined as follows [Sain 1994]:

$$\hat{f}_{AKDE}(x) = \frac{1}{n}\sum_{i=1}^{n} K_{H(z_i)}(x - z_i), \quad H(z_i) \propto f(z_i)^{-1}, \tag{3}$$

where $H(z_i)$ is a bandwidth function that is inversely proportional to the true density $f(z_i)$.

In essence, the AKDE increases its learning capacity (via smaller bandwidth) in regions of high density where the local features are likely to originate from the true distribution. This adaptive scheme enables the AKDE to provide superior estimation accuracy over the standard KDE. Although the AKDE can produce higher estimation quality, its computational cost ($O(n^2)$) far exceeds that of the standard KDE ($O(n)$). In the AKDE, the bandwidth $H(\cdot)$ is computed from the true density $f(\cdot)$. Because the true density is unknown, a pilot function is defined to provide an estimate for $f(\cdot)$. Generally, the pilot function is modeled by the standard KDE. This choice implies that evaluating $H(\cdot)$ is an $O(n)$ process. Therefore, computing a query under the AKDE is an $O(n^2)$ operation since $(H\cdot)$ is computed at least once for each sample point. Because data streams are fast, mutable, and potentially unbounded, applied mining techniques should heed the following constraints: employ fixed memory space and perform at most a linear order scan of the data [Babcock et al. 2002; Garofalakis et al. 2002; O'Callaghan et al. 2002]. Hence, the AKDE approach cannot be directly applied to data streams because it clearly infringes on the linear pass restriction.

3. RELATED WORK

The following section surveys existing KDE algorithms under two broad application categories: offline and online analysis.

3.1. KDE for Offline Analysis

Most of the prior works in KDE were initiated from the perspective of *offline* analysis. Methods of this class assume that data are persistent with no or slow updates. This assumption is not applicable to the *online* setting, where updates are rapid and continuous. Additionally, the computational constraints of online techniques are far more stringent than their offline counterparts. For example, online methods have limited space and can only perform at most a linear pass on the data. Due to the assumption on data persistence and the lack of strict enforcement on computational costs, offline estimators cannot, in general, be directly applied to the online environment.

The following provides a summary of offline KDEs. Offline estimators can be categorized in terms of their bandwidth selection strategy: Cross-Validation (CV) and Plug-In (PI) [Heidenreich et al. 2010; Jones et al. 1996; Loader 1999; Turlach 1993]. Popular examples of CV-based approaches include Integrated Squared Error (ISE) minimization of the leave-one-out strategy used in Bowman [1984] and Rudemo [1982]. By design, CV methods are sensitive to sampling variations and therefore tend to produce undersmoothed estimates and result in poor performance. However, it has been shown that the least squares CV estimator can obtain the optimal convergence rate of $O(n^{-4/5})$ [Stone 1984].

The PI methods mitigate the issues of CV approaches by offering bandwidths that generate smoother estimates and possess faster convergence rates [Heidenreich et al. 2010]. At the core, PI-based approaches enforce some prior knowledge of the higher order pilot estimate in their error minimization criterion. Hence, the estimation quality of the PI-based approaches depends on the assumed distributional form of the pilot estimates. Some well-known PI approaches are found in Hall and Marron [1987], Hall et al. [1991], and Sheather and Jones [1991]. As pointed out by Loader, because both CV- and PI-based methods employ a single global bandwidth, they suffer from an inability to properly estimate global and local features simultaneously [Loader 1999].

Local likelihood density estimation is an approach that can provide effective estimation of both global and local features [Hjort and Jones 1996; Loader 1996]. For a given query point x, the approach fits a polynomial curve of the log density around a local neighborhood of x via a kernel-smoothed maximum likelihood estimate. Varying bandwidth values can be employed to enhance density estimates of both global and local features [Loader 1996]. However, the computational cost of applying varying bandwidth is expensive. To compute the local likelihood density estimate of a single bandwidth requires a summation of the kernel-weighted samples and solving for a system of equations. Determining the solution involves numerical calculation of integrals or, in the case of Gaussian kernel, a closed-form solution can be derived but with degraded estimates [Loader 1996]. Hence, applying varying bandwidths to the local likelihood density estimator would further burden the total computational cost.

Due to the potentially high computational costs of the just described KDE methods (e.g., CV and local likelihood), a number of computationally efficient techniques have been proposed. However, these approaches assume a working environment conducive to offline analysis and therefore cannot be readily applied to an online setting. The following highlights some prior studies for efficiently computing KDEs within an offline environment. Zhang et al. proposed an algorithm to maintain a space-efficient KDE by using *CF-tree* [Zhang et al. 1996] and *CF-Kernels* [Zhang et al. 1999]. However, the method employs a global bandwidth in *CF-Kernels* that can lead to oversmoothing and loss of local density information. Gray et al. proposed a kernel space partitioning technique utilizing a *KD-Tree* and bounded support kernels to offline datasets [Gray and Moore 2003]. The *KD-Tree* reduces computations by pruning kernels that do not contribute to the density query. These KDE computational methods assume that data samples are persistent or, at worst, undergo slow updates, such as in the *CF-tree*. However, in the data stream setting, updates are continuous, data are unbounded, and sample probability distributions can evolve rapidly. Hence, different computational models are required to efficiently and effectively estimate PDFs in data streams. In the next section, more recent innovations in online KDE methods are described.

3.2. KDE for Online Analysis

Recently, some works have attempted to address the issues surrounding the management of online stream-based KDEs. The techniques employed by these works can be classified as sample-based, grid-based, and cluster-based.

3.2.1. Sample-Based and Grid-Based KDE. Sample-based KDE employs a subsampling methodology to reduce the total sample size. It provides an efficient management strategy that gives a consistent throughput performance for any given dataset. Subramaniam et al. proposed an outlier detection algorithm for sensor networks by modeling the probabilistic densities of node observations with sample-based KDEs [Subramaniam et al. 2006]. A global bandwidth based on the Scott's Rule is applied to the KDEs. Wegman et al. employed an online KDE to analyze the behavior of Internet traffic [Wegman and Marchette 2003]. They suggested the use of a sequence-based and exponentially aging sliding window to accommodate a fixed storage environment. To derive estimates, a global bandwidth KDE is utilized. Grid-based KDE generates a uniformly spaced and discretized representation of the sample points. Its sample processing throughput varies between datasets, but can provide improved estimation quality due to its ability to capture aggregated information. Aggarwal proposed a framework to model the structural evolution of data streams by using a grid-based KDE [Aggarwal 2003]. The samples are summarized in a multidimensional grid where the Scott's Rule global bandwidth is employed within each dimension. Concepts of forward and reverse density profiles are introduced to discover the occurrences of concept drifts.

3.2.2. Cluster-Based KDE. Cluster-based KDE performs kernel merging to maintain fixed-size storage. Due to the merging of kernels, the processing throughput is heavily dependent on the data's characteristics, and its throughput can be quite low for continuous-value data [Boedihardjo et al. 2008; Heinz and Seeger 2008]. As a result, the cluster-based estimator does not possess the stability in throughput performance as the sample-based approaches, but it can achieve much higher estimation accuracy. Zhou et al. introduced the *M-Kernel*, a cluster-based KDE maintenance strategy that performs numerically based kernel mergers under a fading window [Zhou et al. 2003]. However, their proposed bandwidth strategy does not guarantee their estimate converges to the true density as the number of samples is increased (i.e., pointwise consistency is not assured). To address the drawbacks of *M-Kernel*, Heinz et al. proposed the *Cluster Kernels,* which employ a constant time pairwise merging technique and a global bandwidth scheme based on the Scott's Rule [Heinz and Seeger 2008]. The *Cluster Kernels* were shown to outperform the *M-Kernel* in terms of estimation quality and throughput performance. To mitigate the problem of the globally assigned bandwidth while ensuring consistency, we previously proposed a cluster-based KDE that supports multiple bandwidth assignments [Boedihardjo et al. 2008].

All of these stream-based KDE approaches depend on a single maintenance strategy: sample-based, grid-based, or cluster-based. Each maintenance strategy possesses its own set of benefits and drawbacks. For example, sample-based KDE achieves stable throughput but can produce lower estimation quality than its grid-based and cluster-based counterparts. The gaps in estimation quality between the various classes of KDEs can be significant (i.e., cluster-based vs. sample-based). As a result, the KDE selection criteria can be biased toward those KDEs that produce high-quality estimates. However, if the qualities of estimates for all classes of KDEs are made sufficiently high, then the suitability of a chosen KDE can be decided on other characteristics (e.g., sample throughput). Hence, this article proposes a generalized algorithmic framework that can be applied to any global bandwidth stream-based KDE to enhance the estimation quality of complex distributions while achieving $O(M)$ query cost.

In addition to these issues, the proposed GLEAM approach differs from our previous KDE method [Boedihardjo et al. 2008] in the following two major aspects:

(1) **GLEAM integrates LR regularization**: In the previous method, the number of local bandwidths is constant, which can lead to poor estimation performance, especially under concept drifts. GLEAM addresses this issue by developing a dynamic local bandwidth generation method (initialized with a large LR number) through a novel online regularization (i.e., reduce local bandwidth count) algorithm.

(2) **In-depth theoretical study of GLEAM**: Rigorous theoretical analyses are provided for GLEAM that include its expected error, convergence rate, and error reduction (with regard to classical global bandwidth KDE). Because GLEAM is a generalization of the previously proposed approach, the theoretical analyses provided in this article are applicable to the prior work. Online histograms have also been developed in the field of database optimization to provide query selectivity estimates and approximate queries [Ioannidis 2003]. Online histograms include dynamic quantiles [Gilbert et al. 2002], equidepth histograms [Gibbons et al. 2002], and V-optimal histograms [Guha et al. 2006]. Due to the histogram's inherent

discontinuities and slower convergence rate, the histogram may not be suited for the tasks of stream analysis [Hardle et al. 2004; Silverman 1986].

4. LOCAL REGION PROPERTIES

This section introduces the concept of the LR-based KDE and provides its estimation properties. Results of the theoretical analyses motivate the application of the LR concept to existing KDEs. In the following section, the estimation quality of the LR- based KDE is derived, analyzed, and evaluated against the standard KDE. In particular, this section focuses on the estimator's pointwise accuracy, global accuracy, convergence rate, and error reduction capacity.

The following provides the definition of the LR-based KDE and assumptions on the kernel functions and bandwidth forms.

Definition 4.1 (*Local Region* (*LR*)). Let S be an i.i.d. sample set for a given time period and D be the ordered set of S. In modeling the data stream, the ordered set D is indexed on a temporal attribute such as arrival time. Let $p = [a, b]$ be a subinterval of D, then an *unrestricted* LR of D is $l = \{z | z \in D \wedge (a \leq z \leq b)\}$. Furthermore, define the lower and upper bounding functions of l as $\Omega(l) = a$ and $\Theta(l) = b$, respectively. Then, the complete set of k LRs of D is $L = \{l_j | \cap^{k}_{i \neq j}((\Omega(l_i), \Theta(l_i)] \cap (\Omega(l_j), \Theta(l_j)]) = \emptyset\}$ and $z \in D \Rightarrow \Omega(l_i) < z \leq \Theta(l_i)$ for exactly one $l_i \in L$. Hence, L is a set of LRs that disjointly partitions the sample set D. Due to this disjoint constraint, each $l_j \in L$ is called a *restricted* LR. Throughout this article, an LR (without a qualifier) refers to the *restricted* LR as just defined.

Definition 4.2 (*LR Center and Radius*). Let l be an LR; then, the center of l is defined as $\text{center}(l) = \frac{1}{|l|} \sum_{z \in l} z$ (i.e., mean of the samples in l). The radius of l is defined as $\text{radius}(l) = \sqrt{\sum_{z \in l} (z - \text{center}(l))^2}$ (i.e., standard deviation of the samples in l).

Definition 4.3 (*LR-based KDE*). For a given i.i.d. sample set D and its corresponding set of LRs L, the LR-based KDE $\hat{f}_{LR}(x)$ of D is defined as follows:

$$\hat{f}_{LR}(x) = \frac{1}{|D|} \sum_{i=1}^{|D|} K_{h_{z_i}}(x - z_i), \tag{4}$$

where $z_i \in D$, $h_{z_i} = H_{LR}(l) \,|\, l \in L \wedge (z_i \cap (\Omega(l), \Theta(l)] \neq \emptyset)$, and $H_{LR}(l)$ is the locally global bandwidth associated with LR l that contains z_i.

Assumptions on Kernel Function: The kernel function $K(\cdot)$ must satisfy the following conditions:

$$K(t) \geq 0, \quad \int K(t)\,dt = 1, \quad \int tK(t)\,dt = 0, \quad \int t^2 K(t)\,dt < \infty \tag{5}$$

Assumptions on the Bandwidth: Each bandwidth h_j is positive and follows Parzen's sufficiency conditions [Parzen 1962]:

$$h_j \to 0 \quad \text{and} \quad |D|h_j \to \infty \quad \text{as} \quad |D| \to \infty \tag{6}$$

The LR-based KDE is an estimator that assigns an identical bandwidth to each data sample within an LR. For example, suppose that D is partitioned into two LRs, l_1 and l_2. The LR-based KDE assigns the bandwidth h_1 to all samples in l_1 and bandwidth h_2 to all samples in l_2. The bandwidths h_1 and h_2 are not necessarily identical because they are entirely a function of l_1 and l_2, respectively. As a result, the maximum number of unique bandwidths for the LR-based KDE is $|L|$. The standard KDE is a special case of the LR-based KDE with $|L| = 1$. Applying Parzen's sufficiency condition for each LR enables the LR-based KDE to achieve asymptotic consistency; that is, as the size of data $|D| \to \infty$, the estimation error approaches 0 (see Sections 4.1–4.3 for detailed analyses and proofs).

As stated in Definition 4.1, LRs provide a total and disjoint partitioning of the data sample set D. The criterion by which to partition D (i.e., construct the LRs) is dependent on the bandwidth function assigned to each LR. Hence, the specification of the LR construction is achieved after establishing the desired condition of the local estimates. For example, if a single LR is chosen to model a unimodal distribution, then the Scott's Rule bandwidth would be an appropriate function to employ within each region. The use of the Scott's Rule would then dictate a particular construction criterion for the LRs.

Justification for Independent and Identical Distribution: Most of the examined stream-based KDEs in this article, including the proposed GLEAM approach, assume that the samples are i.i.d. *conditioned on the time window* [Heinz and Seeger 2008; Zhou et al. 2003]. Because the sliding time window constrains the model to the most recent samples, it is reasonable to assume that these samples originate from the same or identical generating process. When distributional changes occur, samples representing the newest distribution are captured, and old samples from the previous distribution are removed by the sliding window. This identical distribution assumption has also been discussed and justified in other data stream works [Domingos and Hulten 2012; Heinz and Seeger 2008]. In many streaming scenarios, the samples are generated independently, as in Automatic Teller Machine (ATM) transactions; hence, it is reasonable to suppose that no correlation exists between the samples. In settings where data correlation exists, it has been shown theoretically and empirically that the asymptotically optimal bandwidth KDE provides high-quality estimates on dependent samples [Hall et al. 1995].

Error analysis of LR-based KDE under a theoretical framework: To evaluate a given LR bandwidth form H_{LR}, a theoretical framework that allows one to analyze the estimation results induced by H_{LR} is required. To that end, this section derives some concrete forms of the estimation error (e.g., mean squared error; MSE), whereby an arbitrary H_{LR} can be applied and its resulting estimator evaluated. Furthermore, the error forms have been selected in a manner that provides a fair comparison with existing results of the standard KDE. The support for such comparative analysis allows one to determine the conditions for which the LR-based KDE obtains enhanced estimates over the standard KDE.

Consistency and convergence rate of the LR-based KDE: The theoretical framework supports the analyses of both pointwise and global accuracy. These estimation aspects are expressed in terms of the bias, variance, MSE, and mean integrated squared error, which are analyzed in Sections 4.1–4.3. The error forms are then used to establish the estimator's asymptotic consistency. In data streams, consistency can ensure that the expected error will decrease as more samples are processed. However, the rate at which the error diminishes is not provided by consistency alone. Hence, the convergence rate of the LR-based KDE is derived and analyzed. Results show that the LR-based KDE's convergence rate is no lower than the standard KDE's. Under certain complex distributions (i.e., those with large values of the integrated density curvature), and based on the results of the asymptotic mean integrated squared error (Sections 4.1–4.3), it is shown that the LR-based KDE can provide a lower expected error than the standard KDE (Section 4.4). These estimation properties of the LR-based KDE motivates the development of GLEAM.

4.1. Point-Wise Accuracy

In this section, the accuracy of the LR-based KDE is analyzed for a given density query point x. In particular, the estimator's bias and variance are derived, and their relationship to data streams is discussed. Subsequently, the derived bias and variance are used to obtain the estimator's MSE.

4.1.1. Bias and Variance. This subsection discusses the role of the bandwidth h_j in the LR-based KDE's bias (Lemma 4.1) and variance (Lemma 4.2) within the data stream setting. It can be observed that reducing h_j lowers the bias but increases the variance, which results in the common Bias-Variance tradeoff [Hastie et al. 2001]. However, because the estimator is applied to data streams, the error contribution of the bias can far outweigh the error contribution of the variance. Analytically, this observation can be justified as follows: suppose the Asymptotic Mean Integrated Squared Error (AMISE) optimal bandwidth (Equation (14) with regard to standard KDE) is applied to each LR, then $\frac{\text{variance}}{\text{bias}} \rightarrow |D|^{-\frac{2}{5}} \rightarrow 0$ as $|D| \rightarrow \infty$ since bias $\propto h_j^2$ and variance $\propto (|D|h_j)^{-1}$. This result also holds when the Scott's Rule is applied to each LR due to its AMISE optimality with regard to a normal reference. Hence, it is a focus of the LR-based KDE to reduce the bias in data stream applications.

LEMMA 4.1 (BIAS OF LR-BASED KDE). *Let $f(x)$ be the PDF of D and h_j be the bandwidth of LR l_j, then the bias of $\hat{f}_{LR}(x)$ is given as follows:*

$$BIAS\left(\hat{f}_{LR}(x)\right) \leq \left(\sum_{j=1}^{|L|} \frac{|l_j| h_j^2}{|D|}\right)\left(\frac{f''(x)}{2}\int s^2 K(s)\,ds\right) + o\left(\sum_{j=1}^{|L|} \frac{|l_j| h_j^2}{|D|}\right). \tag{7}$$

PROOF. To prove the bias of $\hat{f}_{LR}(x)$, the bias is initially proved with $|L| = 2$ and subsequently generalized to $|L| \in \mathbb{Z}^+$.

The proof begins with the definition of bias, which is given as follows:

$$BIAS\left(\hat{f}_{LR}(x)\right) = E\left[\hat{f}_{LR}(x) - f(x)\right] = E\left[\hat{f}_{LR}(x)\right] - f(x).$$

Suppose that $L = \{^{l_1 : z_i | 1 \leq i \leq m}_{l_2 : z_i | m+1 \leq i \leq |D|}$ and $z_i \leq z_{i+1}$; then, by the definition of $\hat{f}_{LR}(x)$, we have the following:

$$\begin{aligned} E\left[\hat{f}_{LR}(x)\right] &= E\left[\sum_{i=1}^{m} \frac{K_{h_1}(x - z_i)}{|D|} + \sum_{i=m+1}^{|D|} \frac{K_{h_2}(x - z_i)}{|D|}\right], \\ &= \frac{m}{|D|} E\left[\frac{1}{h_1} K\left(\frac{x - \mathbf{Z_1}}{h_1}\right)\right] + \frac{|D| - m}{|D|} E\left[\frac{1}{h_2} K\left(\frac{x - \mathbf{Z_2}}{h_2}\right)\right], \end{aligned}$$

where $E[\frac{1}{h_1}K(\frac{x-\mathbf{Z_1}}{h_1})] = \int_{l_1} \frac{1}{h_1}K(\frac{x-z_i}{h_1})f(z_i)dz \leq \int \frac{1}{h_1}K(\frac{x-z_i}{h_1})f(z_i)dz = E[\frac{1}{h_1}K(\frac{x-\mathbf{Z}}{h_1})]$ and $E[\frac{1}{h_2}K(\frac{x-\mathbf{Z_2}}{h_2})] = \int_{l_2} \frac{1}{h_2}K(\frac{x-\boldsymbol{z_i}}{h_2})f(z_i)dz \leq \int \frac{1}{h_2}K(\frac{x-\boldsymbol{z_i}}{h_2})f(z_i)dz = E[\frac{1}{h_2}K(\frac{x-\mathbf{Z}}{h_2})]$

$$\begin{aligned} E\left[\hat{f}_{LR}(x)\right] &\leq \frac{m}{|D|} E\left[\frac{1}{h_1} K\left(\frac{x - \mathbf{Z}}{h_1}\right)\right] + \frac{|D| - m}{|D|} E\left[\frac{1}{h_2} K\left(\frac{x - \mathbf{Z}}{h_2}\right)\right] \\ &= \frac{m}{|D|} \int \frac{1}{h_1} K\left(\frac{x - z}{h_1}\right) f(z)\,dz + \frac{|D| - m}{|D|} \int \frac{1}{h_2} K\left(\frac{x - z}{h_2}\right) f(z)\,dz. \end{aligned} \tag{8}$$

Let $s_1 = \frac{z-x}{h_1}$ and $s_2 = \frac{z-x}{h_2}$ and derive $\frac{dz}{ds_1}$ and $\frac{dz}{ds_2}$; the following expression is obtained:

$$E\left[\hat{f}_{LR}(x)\right] \leq \frac{m}{|D|} \int K(s_1)\, f(x + s_1 h_1)\, ds_1 + \frac{|D| - m}{|D|} \int K(s_2)\, f(x + s_2 h_2)\, ds_2.$$

Using the second-order Taylor expansion for $f(\cdot)$ and substituting the result into this expression, the following LR-based KDE bias is derived:

$$E\left[\hat{f}_{LR}(x)\right] - f(x) \leq \left(\frac{m}{|D|}h_1^2 + \frac{|D|-m}{|D|}h_2^2\right)\left(\frac{f''(x)}{2}\int s^2 K(s)\,ds\right) + o\left(\frac{m}{|D|}h_1^2 + \frac{|D|-m}{|D|}h_2^2\right).$$

The additive composition of Equation (8) for $|L| = k$ is

$$\frac{m_1}{|D|}\int \frac{1}{h_1}K\left(\frac{x-z}{h_1}\right) f(z)\,dz + \cdots + \frac{m_k}{|D|}\int \frac{1}{h_2}K\left(\frac{x-z}{h_2}\right) f(z)\,dz,$$

where $\sum_{i=1}^{k} m_i = |D|$. Using this generalized form, the expression becomes the final bias expression shown in Equation (7). □

LEMMA 4.2 (VARIANCE OF LR-BASED KDE). *Let $f(x)$ be the PDF of D and h_j be the bandwidth of LR set l_j; then, the variance of $\hat{f}_{LR}(x)$ is given as follows:*

$$VAR\left(\hat{f}_{LR}(x)\right) \leq \left(\sum_{j=1}^{|L|}\frac{|l_j|}{|D|^2 h_j}\right)\left(f(x)\int K(s)^2 ds\right) + o\left(\sum_{j=1}^{|L|}\frac{|l_j|}{|D|^2 h_j}\right). \qquad (9)$$

PROOF. The variance of the LR KDE is initially derived with $|L| = 2$ then generalized to $|L| \in \mathbb{Z}^+$.

Let $L = \{ \substack{l_1 : z_i | 1 \leq i \leq m \\ l_2 : z_i | m+1 \leq i \leq |D|}$ and $z_i \leq z_{i+1}$ then by the definition of $\hat{f}_{LR}(x)$ we have the following:

$$\begin{aligned} VAR\left(\hat{f}_{LR}(x)\right) &= VAR\left(\sum_{i=1}^{m}\frac{K_{h_1}(x-z_i)}{|D|} + \sum_{i=m+1}^{|D|}\frac{K_{h_2}(x-z_i)}{|D|}\right) \\ &= \frac{1}{|D|^2}VAR\left(\sum_{i=1}^{m}K_{h_1}(x-z_i)\right) + \frac{1}{|D|^2}VAR\left(\sum_{i=m+1}^{|D|}K_{h_2}(x-z_i)\right) \\ &= \frac{m}{|D|^2}VAR\left(K_{h_1}(x-\boldsymbol{Z_1})\right) + \frac{|D|-m}{|D|^2}VAR\left(K_{h_2}(x-\boldsymbol{Z_2})\right) \end{aligned}$$

$$\begin{aligned} VAR\left(\hat{f}_{LR}(x)\right) &\leq \frac{m}{|D|^2}\left(VAR\left(K_{h_1}(x-\boldsymbol{Z})\right) + E\left[K_{h_1}(x-\boldsymbol{Z})\right]^2\right) \\ &\quad + \frac{|D|-m}{|D|^2}\left(VAR\left(K_{h_2}(x-\boldsymbol{Z})\right) + E\left[K_{h_2}(x-\boldsymbol{Z})\right]^2\right). \end{aligned} \qquad (10)$$

Applying the Taylor series expansion, the following variance expression is obtained:

$$VAR\left(\hat{f}_{LR}(x)\right) \leq \left(\frac{m}{|D|^2 h_1} + \frac{|D|-m}{|D|^2 h_2}\right) f(x)\int K(s)^2\,ds + o\left(\frac{m}{|D|^2 h_1} + \frac{|D|-m}{|D|^2 h_2}\right).$$

The additive composition of Equation (10) for $|L| = k$ is $\frac{m_1}{|D|^2}VAR(K_{h_1}(x-\boldsymbol{Z})) + \cdots + \frac{m_k}{|D|^2}VAR(K_{h_2}(x-\boldsymbol{Z}))$ where $\sum_{i=1}^{k} m_i = |D|$. Using this generalized form, the expression becomes the final variance expression shown in Equation (9). □

4.1.2. Mean Squared Error (MSE). The MSE of the LR-based KDE is provided in Lemma 4.3. Using the MSE, Theorem 4.1 shows that the pointwise estimate is L_2 consistent [Lehmann 1998]. Consistency assures that the LR-based KDE will converge to the true density as the sample size approaches infinity (i.e., $|D| \to \infty$). This property can be observed in Equation (11), where the MSE of the estimate at x approaches zero as the sample size $|D|$ diverges to infinity. For data streams, this property is especially important because the large and increasing number of samples is guaranteed (assuming that the maximum rate of decrease of any h_j is bounded by $(|D|+1)^{-1}$) to reduce the LR-based KDE's mean pointwise error.

LEMMA 4.3 (MEAN SQUARED ERROR (MSE) OF THE LR-BASED KDE). *Let $f(x)$ be the PDF of D, h_j be the bandwidth of LR set l_j, and select h_j such that it satisfies the conditions of Equation (6); then, the MSE of $\hat{f}_{LR}(x)$ is given as follows:*

$$MSE\left(\hat{f}_{LR}(x)\right) \leq \left(\sum_{j=1}^{|L|} \frac{|l_i|\,h_j^2}{|D|}\right)^2 \left(\int s^2 K(s)\,ds\right)^2 \frac{f''(x)^2}{4} + \left(\sum_{j=1}^{|L|} \frac{|l_j|}{|D|^2\,h_j}\right)\left(\int K(s)^2\,ds\right) f(x)$$
$$+\, o\left(\sum_{j=1}^{|L|} \frac{|l_i|\,h_j^2}{|D|}\right) + o\left(\sum_{j=1}^{|L|} \frac{|l_j|}{|D|^2 h_j}\right). \tag{11}$$

PROOF. Recall that

$$MSE\left(\hat{f}\right) = BIAS(\hat{f})^2 + VAR(\hat{f}).$$

Substituting Equation (7) (bias) and Equation (9) (variance) into this expression, the MSE of $\hat{f}_{LR}(x)$ becomes the following:

$$\begin{aligned} MSE\left(\hat{f}_{LR}(x)\right) \;\leq\; & \left(\int s^2 K(s)\,ds\right)^2 \frac{f''(x)^2}{4} + \left(\sum_{j=1}^{|L|} \frac{|l_j|}{|D|^2\,h_j}\right)\left(\int K(s)^2\,ds\right) f(x) \\ & + o\left(\sum_{j=1}^{|L|} \frac{|l_j|\,h_j^2}{|D|}\right) + o\left(\sum_{j=1}^{|L|} \frac{|l_j|}{|D|^2 h_j}\right) + o\left(\sum_{j=1}^{|L|} \frac{|l_j|\,h_j^2}{|D|}\right)^2 \end{aligned}$$

Because each h_j fulfills Parzen's sufficiency conditions (Equation (6)), the MSE in Equation (11) is obtained. □

THEOREM 4.1 (MSE-CONSISTENCY OF LR-BASED KDE). *Given the conditions of the kernel function (Equation (5)) and assumptions on the bandwidth (Equation (6)), the LR-based KDE is MSE (pointwise) consistent.*

PROOF. To prove MSE consistency, we show that the LR-based KDE's MSE (Equation (11)) approaches 0 as $|D| \to \infty$.

In the following, it is shown that $\sum_{j=1}^{|L|} \frac{|l_j| h_j^2}{|D|}$ and $\sum_{j=1}^{|L|} \frac{|l_j|}{|D|^2 h_j}$ converge to 0 as $|D| \to \infty$:

$$\begin{aligned} \sum_{j=1}^{|L|} \frac{|l_j| h_j^2}{|D|} \;&\leq\; \sum_{j=1}^{|L|} \frac{\max\left|l_{1\leq i\leq |L|}\right| \cdot \max\left|h^2_{1\leq i\leq |L|}\right|}{|D|} \\ &=\; \frac{|L| \cdot \max\left|l_{1\leq i\leq |L|}\right|}{|D|} \cdot \max\left|h^2_{1\leq i\leq |L|}\right| \\ &=\; k \cdot \max\left|h^2_{1\leq i\leq |L|}\right| \to 0 \quad \text{as} \quad |D| \to \infty. \end{aligned}$$

and

$$\sum_{j=1}^{|L|} \frac{|l_j|}{|D|^2 h_j} \leq \sum_{j=1}^{|L|} \frac{1}{|D|\, h_j} \leq |L| \frac{1}{|D| \cdot \min\left(h_{1 \leq i \leq |L|}\right)} \to 0 \quad \text{as} \quad |D| \to \infty.$$

From the kernel conditions, we have:

$$\left(\int s^2 K(s)\, ds\right)^2 \frac{f''(x)^2}{4} < \infty \, \text{and} \left(\int k(s)^2\, ds\right) f(x) < \infty.$$

Therefore $MSE(\hat{f}_{LR}(x)) \to 0$ as $|D| \to \infty$. □

4.2. Global Accuracy

To provide an understanding of the estimator's global accuracy, the analysis of the LR-based KDE is performed on the *entire* support of the density $f(\cdot)$. The global error is defined as the *cumulative* pointwise error ($\int MSE(\hat{f}_{LR}(x))dx$) in the complete domain space. In Theorem 4.2, it is shown that the cumulative pointwise error (MISE) of the LR-based KDE converges to zero; therefore, the LR-based KDE is L_2 consistent within the support of the density. Notice that the $||f''||_2^2$ term in Equation (12) describes the aggregated rate of fluctuations in the density $f(x)$ (i.e., $||f''||_2^2$ quantifies the overall complexity of the density). It will be shown in Section 4.4 that the LR-based KDE can reduce the error generated by $||f''||_2^2$ via a reduction in the integrated squared bias weight $\sum_{j=1}^{|L|} \frac{|l_j| h_j^2}{|D|}$. Similar to the pointwise estimate, reducing the integrated squared bias weight produces an increase in the integrated variance weight $\sum_{j=1}^{|L|} \frac{|l_j|}{|D|^2 h_i}$. However, as discussed in Section 4.1, the effective contribution of the integrated variance is relatively small due to the large sample size of the data stream.

LEMMA 4.4 (ASYMPTOTIC MEAN INTEGRATED SQUARED ERROR (AMISE) OF THE LR-BASED KDE). *Let $f(x)$ be the PDF of D, h_j be the bandwidth of LR set l_j, and select h_j such that it satisfies the conditions of Equation (6), then the AMISE of $\hat{f}_{LR}(x)$ is given as follows:*

$$AMISE\left(\hat{f}_{LR}(x)\right) \leq \left(\sum_{j=1}^{|L|} \frac{|l_j|\, h_j^2}{|D|}\right)^2 \frac{\upsilon_2(K)^2}{4} ||f''||_2^2 + \left(\sum_{j=1}^{|L|} \frac{|l_j|}{|D|^2 h_i}\right) ||K||_2^2, \tag{12}$$

where $\upsilon_2(K) = \int s^2 K(s) ds$, $||K||_2^2 = \int K(s)^2 ds$, and $||f''||_2^2 = \int f''(x)^2\, dx$.

PROOF. Notice that

$$MISE\left(\hat{f}_{LR}(x)\right) = \int MSE\left(\hat{f}_{LR}(x)\right) dx.$$

Substituting the MSE (Equation (11)) into this MISE expression gives the following LR-based KDE MISE:

$$\begin{aligned} MISE\left(\hat{f}_{h(i)}(x)\right) \leq & \left(\sum_{j=1}^{|L|} \frac{|l_j|\, h_j^2}{|D|}\right)^2 \frac{\upsilon_2(K)^2}{4} ||f''||_2^2 + \left(\sum_{j=1}^{|L|} \frac{|l_j|}{|D|^2 h_i}\right) ||K||_2^2 \\ & + o\left(\sum_{i=1}^{|L|} \frac{|l_j|\, h_j^2}{|D|}\right) + o\left(\sum_{i=1}^{|L|} \frac{|l_j|}{|D|^2 h_j}\right), \end{aligned} \tag{13}$$

where $\upsilon_2(K) = \int s^2 K(s) ds$, $||K||_2^2 = \int K(s)^2 ds$, and $||f''||_2^2 = \int f''(x)^2\, dx$.

Let $|D| \to \infty$ and, since each bandwidth h_j fulfills Parzen's sufficiency condition (Equation (6)), the AMISE of the LR-based KDE simplifies to Equation (12). □

THEOREM 4.2 (MISE CONSISTENCY OF LR-BASED KDE). *Given the conditions of the kernel function (Equation (5)) and assumptions on the bandwidth (Equation (6)), the LR-based KDE is MISE (globally) consistent.*

PROOF. From the result of the MSE consistency it is shown that

$$\sum_{j=1}^{|L|} \frac{|l_j| h_j^2}{|D|} \quad \text{and} \quad \sum_{j=1}^{|L|} \frac{|l_j|}{|D|^2 h_j} \text{ converge to 0 as } |D| \to \infty.$$

Furthermore, from the kernel conditions we have:

$$\frac{\upsilon_2(K)^2}{4} ||f''||_2^2 < \infty \quad \text{and} \quad ||K||_2^2 < \infty,$$

where $\upsilon_2(K) = \int s^2 K(s) ds$, $||K||_2^2 = \int K(s)^2 ds$, and $||f''||_2^2 = \int f''(x)^2 dx$.

Therefore, by applying the limits above to the MISE expression (Equation (13)), we have:

$$MISE\left(\hat{f}_{LR}(x)\right) \to 0 \quad \text{as} \quad |D| \to \infty. \quad \square$$

4.3. Convergence Rate

The convergence rate of a KDE is dependent on the selected form of the bandwidth. In this subsection, the AMISE optimal bandwidth of the KDE is applied to the LR-based KDE, and its convergence rate is compared against the standard KDE (Equation (1)). The AMISE optimal bandwidth for the standard KDE is transformed into an LR-based KDE compatible form as follows:

$$h_j = \left(\frac{||K||_2^2}{||f_j''||_2^2 u_2(K)^2 |l_j|} \right)^{\frac{1}{5}}, \tag{14}$$

where $\upsilon_2(K) = \int s^2 K(s) ds$, $||K||_2^2 = \int K(s)^2 ds$, and $||f_i''||_2^2 = \int f_i''(x)^2 dx$ (i.e., squared L_2 norm of the density curvature of l_i).

Substituting the bandwidth shown here into Equation (12) gives the following bandwidth form-specific AMISE:

$$\begin{aligned} AMISE\left(\hat{f}_{LR}(x)\right) \leq\; & \frac{||f''||_2^2 u_2(K)^2}{4|D|^2} \left(\sum_{i=1}^{|L|} \left(\frac{|l_i|^{\frac{3}{2}} ||K||_2^2}{||f_i''||_2^2 u_2(K)} \right)^{\frac{2}{5}} \right)^2 \\ & + \frac{||K||_2^2}{|D|^2} \sum_{i=1}^{|L|} \left(\frac{|l_i|^6 ||f_i''||_2^2 u_2(K)^2}{||K||_2^2} \right)^{\frac{1}{5}} = O\left(|D|^{\frac{-4}{5}}\right). \end{aligned} \tag{15}$$

In comparison, the AMISE of the standard KDE with the optimal bandwidth is as follows:

$$AMISE\left(\hat{f}_{KDE}(x)\right) = \frac{5}{4}\left(||K||_2^2\right)^{\frac{4}{5}} \left(u_2(K)||f''||_2^2\right)^{\frac{2}{5}} |D|^{\frac{-4}{5}} = O\left(|D|^{\frac{-4}{5}}\right). \tag{16}$$

From Equations (15) and (16), the convergence rates of the LR-based KDE and the global bandwidth approaches are identical. This result is not surprising because the number of LRs is constant and orthogonal to the sample size $|D|$. We also note that the optimality of the global bandwidth KDE does not necessarily imply the optimality of the LR-based KDE. The *true* AMISE-optimal LR-based KDE bandwidth can be produced by utilizing the gradient of Equation (12) and determining h_j, which produces zero gradient. The steps to attaining a closed-form solution involve the determination of quintic roots. However, it is guaranteed that the AMISE optimal bandwidth for the LR-based KDE will provide a convergence rate that is no worse than $O(|D|^{\frac{-4}{5}})$.

4.4. LR-based KDE Error Reduction over the Standard KDE

This subsection compares and analyzes the AMISE of the LR-based KDE and standard KDE under the commonly used Scott's Rule bandwidth. The Scott's Rule bandwidth provides oversmoothed estimates of complex structures. However, it can give accurate estimates of simple features (e.g., unimodal density) and is amenable to efficient implementations [Scott 1992; Silverman 1986]. If each LR is tasked to capture the simple features of the density, then applying the Scott's Rule bandwidth to each LR can result in improved estimation quality over the standard KDE. In the following theorem, we describe the conditions for which the LR-based KDE provides lower AMISE than the standard KDE under this specific application of the Scott's Rule bandwidth.

THEOREM 4.3 (AMISE REDUCTION OF LR-BASED KDE). *Let the AMISE difference between the standard KDE and LR-based KDE be defined as* $Z(\hat{f}_{KDE}, \hat{f}_{LR}) = AMISE(\hat{f}_{KDE}(x)) - AMISE(\hat{f}_{LR}(x))$ *and suppose that the Scott's Rule bandwidth is applied to the standard KDE and to each LR of the LR-based KDE (as shown in Equation (14)) and* $|D| > 0$, *then*

$$Z\left(\hat{f}_{KDE}, \hat{f}_{LR}\right) = AMISE\left(\hat{f}_{KDE}(x)\right) - AMISE\left(\hat{f}_{LR}(x)\right) > 0$$

$$\Leftrightarrow \left(|D|^{\frac{6}{5}}\sigma_D^4 - \left(\sum_{i=1}^{|L|}|l_j|^{\frac{3}{5}}\sigma_j^2\right)^2\right)||f''||_2^2 > \alpha_2(K)\left(\sum_{i=1}^{|L|}\frac{|l_j|^{\frac{6}{5}}}{\sigma_j} - \frac{|D|^{\frac{6}{5}}}{\sigma_D}\right), \tag{17}$$

where σ_D *is the standard deviation of D,* σ_j *is the standard deviation of* l_j, $\alpha_2(K) = \frac{4||K||_2^2}{C^5 u_2(K)^2}$, *and C is a constant that is dependent on the kernel function K (see Equation (2)).*

PROOF. Apply the Scott's Rule (Equation (2)) to both the standard KDE and LR-based KDE to obtain the following:

$$Z\left(\hat{f}_{KDE}, \hat{f}_{LR}\right) = \frac{C^4 u_2(K)^2}{4|D|^4}\left(\left(|D|^{\frac{3}{5}}\sigma_D^2\right)^2 - \left(\sum_{i=1}^{|L|}|l_j|^{\frac{3}{5}}\sigma_j^2\right)^2\right)||f''||_2^2 + \frac{||K||_2^2}{C|D|^2}\left(\frac{|D|^{\frac{6}{5}}}{\sigma_D} - \sum_{i=1}^{|L|}\frac{|l_j|^{\frac{6}{5}}}{\sigma_j}\right)$$

$$\Rightarrow \left(|D|^{\frac{6}{5}}\sigma_D^4 - \left(\sum_{i=1}^{|L|}|l_j|^{\frac{3}{5}}\sigma_j^2\right)^2\right)||f''||_2^2 = Z\left(\hat{f}_{KDE}, \hat{f}_{LR}\right)\alpha_1(K, D) + \alpha_2(K)\left(\sum_{i=1}^{|L|}\frac{|l_j|^{\frac{6}{5}}}{\sigma_j} - \frac{|D|^{\frac{6}{5}}}{\sigma_D}\right),$$

where σ_D is the standard deviation of D, σ_j is the standard deviation of l_j, $\alpha_1(K, D) = \frac{4|D|^2}{C^4 u_2(K)^2}$, $\alpha_2(K) = \frac{4||K||_2^2}{C^5 u_2(K)^2}$, and C is a constant that is dependent on the kernel function K as shown in Equation (2).

Suppose $Z(\hat{f}_{KDE}, \hat{f}_{LR}) > 0$ (i.e., the AMISE of the LR-based KDE is lower than the standard KDE), then we have $Z(\hat{f}_{KDE}, \hat{f}_{LR})\alpha_1(K, D) > 0$ because $\alpha_1(K, D) = \frac{4|D|^2}{C^4 u_2(K)^2} > 0$. Hence,

$$\left(|D|^{\frac{6}{5}}\sigma_D^4 - \left(\sum_{i=1}^{|L|} |l_j|^{\frac{3}{5}}\sigma_j^2\right)^2\right) ||f''||_2^2 > \alpha_2(K)\left(\sum_{i=1}^{|L|} \frac{|l_j|^{\frac{6}{5}}}{\sigma_j} - \frac{|D|^{\frac{6}{5}}}{\sigma_D}\right)$$

It is straightforward to show that if the this condition holds, then $Z(\hat{f}_{KDE}, \hat{f}_{LR}) > 0$ since $\alpha_1(K, D) > 0$ (i.e., converse relationship is true). Therefore,

$$Z\left(\hat{f}_{KDE}, \hat{f}_{LR}\right) > 0 \Leftrightarrow \left(|D|^{\frac{6}{5}}\sigma_D^4 - \left(\sum_{i=1}^{|L|} |l_j|^{\frac{3}{5}}\sigma_j^2\right)^2\right) ||f''||_2^2 > \alpha_2(K)\left(\sum_{i=1}^{|L|} \frac{|l_j|^{\frac{6}{5}}}{\sigma_j} - \frac{|D|^{\frac{6}{5}}}{\sigma_D}\right). \quad \square$$

Theorem 4.3 shows that if the parameters of the LR-based KDE satisfies Equation (17), then it is guaranteed that the LR-based KDE's AMISE will be lower than the standard KDE. The form of the expression in Equation (17) demonstrates the relations between the curvature $||f''||_2^2$ (i.e., complexity of the PDF), the AMISE difference $Z(\hat{f}_{KDE}, \hat{f}_{LR})$, and the parameters of the KDEs in a somewhat complicated manner. To simplify these relations, we investigate the conditions for which $(|D|^{\frac{6}{5}}\sigma_D^4 - (\sum_{i=1}^{|L|} |l_j|^{\frac{3}{5}}\sigma_j^2)^2) > 0$. Because $\sqrt{|D|^{\frac{6}{5}}\sigma_D^4} > 0$ and $\sum_{i=1}^{|L|} \left|l_j\right|^{\frac{3}{5}} \sigma_j^2 > 0$, we have

$$\left(|D|^{\frac{6}{5}}\sigma_D^4 - \left(\sum_{i=1}^{|L|} \left|l_j\right|^{\frac{3}{5}}\sigma_j^2\right)^2\right) > 0 \Leftrightarrow \left(|D|^{\frac{3}{5}}\sigma_D^2 - \sum_{i=1}^{|L|} \left|l_j\right|^{\frac{3}{5}}\sigma_j^2\right) > 0$$

Furthermore,

$$\begin{aligned}\left(|D|^{\frac{3}{5}}\sigma_D^2 - \sum_{i=1}^{|L|} \left|l_j\right|^{\frac{3}{5}}\sigma_j^2\right) &\geq \left(|D|^{\frac{3}{5}}\sigma_D^2 - \sum_{i=1}^{|L|} \left|l_j\right|^{\frac{3}{5}}\left(\max\{\sigma_1, \ldots, \sigma_j, \ldots, \sigma_{|L|}\}^2\right)\right) \\ &= |D|^{\frac{3}{5}} - \sum_{i=1}^{|L|} |l_j|^{\frac{3}{5}}\frac{\max\{\sigma_1, \ldots, \sigma_j, \ldots, \sigma_{|L|}\}^2}{\sigma_D^2}.\end{aligned}$$

The lower bound $|D|^{\frac{3}{5}} - \sum_{i=1}^{|L|} |l_j|^{\frac{3}{5}}\frac{\max\{\sigma_1,\ldots,\sigma_j,\ldots,\sigma_{|L|}\}^2}{\sigma_D^2}$ is minimized when each $\left|l_j\right| = 1$. This condition is achieved when $|L| = |D|$; however, in practice, $|L| < |D|$. Hence, we consider this minimum to be the worst-case scenario for the lower bound expression. Under this worst-case scenario, it is guaranteed that the lower bound $|D|^{\frac{3}{5}} - \sum_{i=1}^{|L|} |l_j|^{\frac{3}{5}}(\frac{\max\{\sigma_1,\ldots,\sigma_j,\ldots,\sigma_{|L|}\}^2}{\sigma_D^2}) > 0$ when $\frac{\max\{\sigma_1,\ldots,\sigma_j,\ldots,\sigma_{|L|}\}}{\sigma_D} < |D|^{\frac{-1}{5}}$. Therefore, if $\frac{\max\{\sigma_1,\ldots,\sigma_j,\ldots,\sigma_{|L|}\}}{\sigma_D} < |D|^{\frac{-1}{5}}$, then $|D|^{\frac{6}{5}}\sigma_D^4 - (\sum_{i=1}^{|L|} |l_j|^{\frac{3}{5}}\sigma_j^2)^2 > 0$ which implies that

$$Z\left(\hat{f}_{KDE}, \hat{f}_{LR}\right) > 0 \Leftrightarrow ||f''||_2^2 > \beta(K, L, D), \tag{18}$$

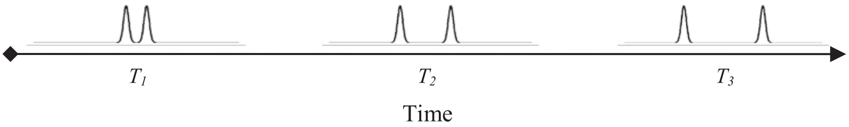


Fig. 1. An example of concept drift due to modal distance shifts.

where $\beta(K, L, D) = \alpha_2(K)(\sum_{i=1}^{|L|} \frac{|l_j|^{\frac{6}{5}}}{\sigma_j} - \frac{|D|^{\frac{6}{5}}}{\sigma_D})/(|D|^{\frac{6}{5}}\sigma_D^4 - (\sum_{i=1}^{|L|} |l_j|^{\frac{3}{5}}\sigma_j^2)^2)$ is called the parameter difference ratio.

Based on this expression, the LR-based KDE can generate lower estimation errors than can the standard KDE when the structural makeup of the true density is complex (i.e., high $||f''||_2^2$ value). This observation can also be implied from Equation (17), where the dominance of $||f''||_2^2$ can lead to the dominance of $(|D|^{\frac{6}{5}}\sigma_D^4 - (\sum_{i=1}^{|L|} |l_j|^{\frac{3}{5}}\sigma_j^2)^2)||f''||_2^2$ over $\alpha_2(K)(\sum_{i=1}^{|L|} \frac{|l_j|^{\frac{6}{5}}}{\sigma_j} - \frac{|D|^{\frac{6}{5}}}{\sigma_D})$, which implies that $AMISE(\hat{f}_{KDE}(x)) - AMISE(\hat{f}_{LR}(x)) > 0$. In short, the higher the complexity of distributional structure (e.g., multimodal), the more likely that the LR-based KDE will generate lower AMISE than the standard KDE.

Another important result of Equation (18) is the implication on the estimation quality under the conditions of concept drifts. In particular, the LR-based KDE can provide stable estimates of systems that undergo modal shifts. Such evolutionary behavior can be exemplified by a binormal density that exhibits changes to its mode center distance (see Figure 1). In this simulated scenario, the density estimates are performed by employing a batch process on the data samples at each of the estimated timestamps. This scenario aims to show the proposed estimator's capacity to adjust to new distributions. Issues regarding the management of streaming data samples (e.g., sample reweighting) are discussed in Section 5. According to the parameter difference ratio $\beta(K, L, D)$ in Equation (18), the LR-based KDE allows for the adaptation to modal shifts and produces (in general) constant error under such distributional mutations. In this scenario, the LR-based KDE's parameters $|l_j|$ and σ_j remain constant, and the standard KDE parameter σ_D increases in proportion to the distance of the mode centers. Furthermore, the aggregated curvature $||f''||_2^2$ is relatively unchanged when the distance between the mode centers exceeds the sum of the mode scales. The increase in mode center distance lowers the parameter difference ratio but retains the aggregated curvature fixed. Hence, the LR-based KDE can provide a fairly consistent AMISE as the mode center distance increases. Assuming in Equation (18) that $||f''||_2^2$ dominates $\alpha_2(K)$, then the standard KDE's error increases as the modal distance becomes larger. Therefore, as the distance between the mode centers increases, the LR-based KDE generates better estimation results than does the standard KDE. This adaptive characteristic of the LR-based KDE is critical because it allows for high-quality estimates under dynamic stream environments.

4.5. Application of LR to Existing KDEs

It can be concluded from Section 4.4 that the LR-based KDE can improve the estimation quality over the global bandwidth KDE for complex densities (i.e., densities with sufficiently large $||f''||_2^2$). The conditions for this result are embedded within the assumption of the AMISE: The samples of D are i.i.d. This assumption enables the application of existing stream-based KDEs that employs i.i.d. assumptions conditioned on a time window [Heinz and Seeger 2008; Zhou et al. 2003]. The significance of the i.i.d. samples is that it guarantees that the sample set accurately represents the distribution

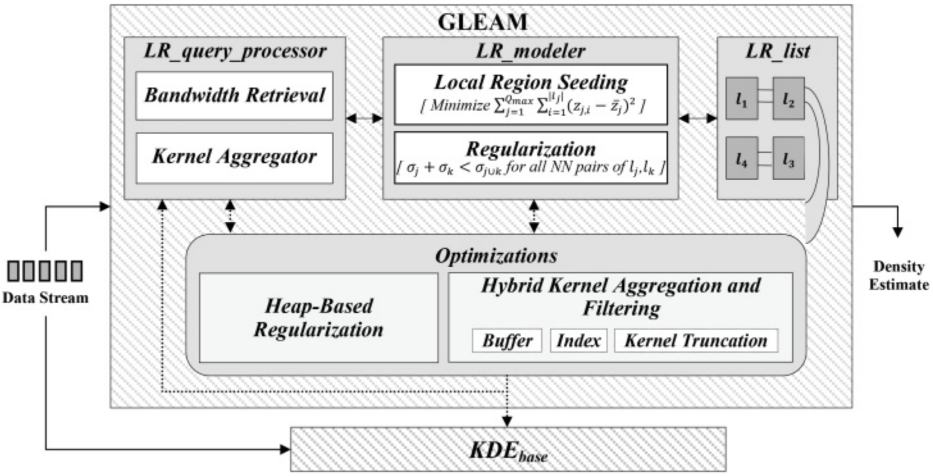

GLEAM
LR_query_processor
Bandwidth Retrieval
Kernel Aggregator
LR_modeler
Local Region Seeding
[Minimize $\sum_{j=1}^{Q_{max}} \sum_{i=1}^{|l_j|} (z_{j,i} - \bar{z}_j)^2$]
Regularization
[$\sigma_j + \sigma_k < \sigma_{j \cup k}$ for all NN pairs of l_j, l_k]
LR_list
l_1
l_2
l_4
l_3
Optimizations
Heap-Based Regularization
Hybrid Kernel Aggregation and Filtering
Buffer
Index
Kernel Truncation
Data Stream
Density Estimate
KDE_{base}

Fig. 2. GLEAM Architecture.

of the original data. Data can be synopsized under various techniques, such as clustered/merged objects, discretized grids, and sub-sampled points. These representations can be regarded as a sampling of the original dataset. Depending on the summarization technique, the represented set can be regarded as i.i.d. samples. This fact implies that the LR bandwidth scheme can be applied to existing stream-based KDEs, given that the representative sampling components accurately model the distribution of the original dataset. However, methods such as gridding may violate the i.i.d. property due to the nonuniform modeling associated with each summarized object. However, such representation errors can often be parameterized and bounded to an arbitrarily small error ε.

5. GLEAM: GENERALIZED LOCAL REGION ALGORITHM

The discussion of Section 4.4 demonstrates that the LR-based KDE can provide improved estimation accuracy over the standard KDE. Furthermore, Section 4.5 shows that the LR scheme can be applied to existing stream-based KDEs to enhance their estimation quality. This characteristic of the LR scheme motivates the development of the GLEAM. GLEAM is an LR bandwidth modeling approach that can be integrated into an existing global bandwidth stream-based KDE. In addition to providing enhanced bandwidths, GLEAM employs efficient strategies to attain time and space costs that are at most $O(M)$. Here, M refers to the number of kernel objects maintained by the existing stream-based KDE, KDE_{base}.

Figure 2 depicts the proposed GLEAM architecture. GLEAM possesses the following three components: *LR_list*, *LR_modeler*, and *LR_query_processor*. *LR_list* maintains a linked list of LRs for the current stream, *LR_modeler* directs the construction of the LRs within *LR_list*, and *LR_query_processor* coordinates and resolves the search for the bandwidths to be assigned to the kernel objects in KDE_{base}. Utilizing these three components, GLEAM performs the following two major tasks: LR construction and density query processing. For LR construction, the *LR_modeler* continuously maintains a set of LRs stored in *LR_list* that reflects the current state of the data stream. For density query processing, *LR_query_processor* performs bandwidth searches for the kernel objects in KDE_{base} and assembles the final density result.

The following provides a short example of the data processing in GLEAM. Each stream sample is forwarded to the KDE_{base} and *LR_modeler*. The KDE_{base} processes the sample following its original approach in maintaining the M representative kernel objects. For the *LR_modeler*, the sample is processed to continuously maintain a set of LRs that describes the current stream. This process of GLEAM is known as "LR construction," which is discussed in Section 5.1. To generate a density query estimate, the *LR_query_processor* module retrieves the relevant LRs and kernel objects from the *LR_modeler*, *LR_list*, and KDE_{base}, and aggregates the kernel contribution values to obtain the final result. The detailed description of the density query algorithm is provided in Section 5.2. Cost analyses and optimizations techniques are provided in Sections 5.3 and 5.4, respectively.

5.1. Local Region Construction with LR_modeler

The LR construction is formulated as an optimization task. The following provides the objective criterion.

Local region construction objective criterion: To construct the LRs, a criterion is proposed based on the application of the Scott's Rule (Equation (2)) to each LR. Imposing this bandwidth results in an error form (see Section 4.4) that estimates the local deviation σ_j for $1 \leq j \leq |L|$. To produce a low parametric difference ratio $\beta(K, L, D)$ relative to the aggregated curvature $||f''||_2^2$ in Equation (18), the following LR assignment heuristic is proposed: each LR is tasked to encapsulate a single mode in order to provide a reasonable balance between the numerator and denominator of $\beta(K, L, D)$. Hence, if each LR is designed to model a unimodal density, then the construction of the LRs should aim to minimize the total Sum Squared Error (SSE). If the SSE minimization objective is employed, then the number of LRs Q must be determined. Because each LR is aimed to capture a unimodal structure, the natural choice for Q is the number of true modes in the density. Verifying against Equation (18), this choice of Q will give a relatively low parametric difference ratio $\beta(K, L, D)$.

To estimate the number of modes within the density, σ_j is used to measure the suitability of the unimodality assumption imposed on LR l_j. Suppose that a representative sample set D of the density is partitioned into r number of LRs constructed via the SSE criterion. Let σ_j and σ_k be the standard deviations of the nearest neighbor pair of LRs l_j and l_k, respectively. Furthermore, define $\sigma_{j\cup k}$ to be the standard deviation of the merged pair $l_j \cup l_k$. The nearest neighbor of l_j is the LR that has a center closest in L_2 distance to the center of l_j. For the univariate case, the nearest neighbor candidates of l_j are the pair of l_j's adjacent LRs. If l_j and l_k are both unimodal, then in the general case $\sigma_{j\cup k} > \sigma_j + \sigma_k$ holds. However, if $\sigma_{j\cup k} \leq \sigma_j + \sigma_k$, then $l_j \cup l_k$ is a more accurate representation of the unimodal structure than l_j and l_k individually. In this case, l_j and l_k should be merged to improve the opportunity of capturing a unimodal structure. If the merging process is performed until each nearest neighbor pair (l_j, l_k) satisfies $\sigma_{j\cup k} > \sigma_j + \sigma_k$, then the resulting number of LRs $s \leq r$ would be a suitable estimate for the number of modes in the sample set D.

Based on this mode estimation strategy, the *LR_modeler* generates the LR in two phases: LR seeding and LR regularization. In LR seeding, the *LR_modeler* continuously constructs and maintains at maximum Q_{max} number of LRs in the *LR_list* where Q_{max} is set to the application's maximum allowable size. In the second phase, the *LR_modeler* regularizes the LRs in *LR_list* by merging those regions that invalidate the unimodality assumption. The result of this regularization phase guarantees that any nearest neighbor pair of LRs (l_j, l_k) satisfies $\sigma_{j\cup k} > \sigma_j + \sigma_k$. The regularized/merged LRs are then used to compute the density estimates.

Phase 1. LR seeding: The process of LR seeding employs a single-scan incremental K-Means clustering on the data stream [Aggarwal et al. 2003]. Let Q_{max} be the maximum number of LRs that can be stored in *LR_list* to provide a fine granular representation of the data stream. Each LR l_j maintains a vector of local statistics $\vec{v}_j = \langle \sum_{i=1}^{|l_j|} z_i^0, \sum_{i=1}^{|l_j|} z_i^1, \sum_{i=1}^{|l_j|} z_i^2 \rangle$ where $z_i \in l_j$. Other required statistics, such as center(l_j) and radius(l_j), can be quickly computed from $\vec{v}_j$. LR addition/subtraction is defined as the componentwise addition/subtraction operation that results in efficient computation of LR merging and splitting.

When a new data sample d arrives, the *LR_modeler* identifies the intersecting l_j in *LR_list* and updates its corresponding $\vec{v}_j$. Here, intersection is defined as follows: d and l_j intersects if and only if $(d \cap [(\text{center}(l_j) - \mu \cdot \text{radius}(l_j)) \cdots (\text{center}(l_j) + \mu \cdot \text{radius}(l_j))]) \neq \emptyset$ and where $\mu \geq 1$. However, if the *LR_modeler* cannot find an intersecting l_j, a new LR is created for d. The construction process continues in this manner until the number of LRs is greater than Q_{max} (i.e., $|LR\_list| > Q_{max}$). If $|LR\_list| > Q_{max}$, the *LR_modeler* merges the two nearest-neighbor LRs that minimize the SSE residual until $|LR\_list| = Q_{max}$. The merging of any two LRs is achieved by performing the addition operation on the corresponding pair of vectors. Hence, the merge operation can be efficiently performed in $O(\dim(\vec{v}))$ time where $\dim(\vec{v})$ is the dimension of $\vec{v}$.

Phase 2. LR regularization: The number of LRs in the *LR_list* can be much larger than the actual number of modes in the density. This condition can cause certain LRs to break the minimum unimodality assumption and artificially deflate the bandwidths (i.e., overfit the data). To resolve this issue, the *LR_modeler* regularizes *LR_list* by agglomerating (at query time) the nearest-neighbor pairs of LRs l_j and l_k that have $\sigma_{j \cup k} \leq \sigma_j + \sigma_k$. The nearest neighbor LR pair is defined as the two LRs with the minimum L_2 distance between their centroids from all other LR pair sets.

To regularize the LRs, the *LR_modeler* employs an incremental merge strategy that combines the nearest-neighbor LRs until no pair of LRs can satisfy the merge condition (i.e., all resulting LR pairs suffices $\sigma_{j \cup k} > \sigma_j + \sigma_k$). In the first step, the LRs of *LR_list* are copied to the regularized LR list L_{reg}. Next, the *LR_modeler* scans L_{reg} to obtain the pair of nearest-neighbor LRs (i.e., has minimum L_2 distance between their centroids) that satisfies the merge condition (i.e., $\sigma_{j \cup k} \leq \sigma_j + \sigma_k$ for LRs l_j and l_k). The scan process takes $O(|L_{reg}|)$ time since only the adjacent pairs of LRs are required for consideration. After obtaining the nearest-neighbor LRs, they are combined to form a new LR to replace the original pair within L_{reg}. As a result of this merging process, $|L_{reg}|$ is reduced by one LR. The algorithm continues to scan and combine the nearest-neighbor merge candidates until there is no pair of LRs that satisfies the merge condition $\sigma_{j \cup k} \leq \sigma_j + \sigma_k$. Hence, at the algorithm's termination, L_{reg} will contain the regularized LRs where $|L_{reg}| \leq |L|$ and each nearest neighbor l_j and l_k fulfills $\sigma_{j \cup k} > \sigma_j + \sigma_k$.

Setting the Q_{max} Parameter: Q_{max} is defined as the upper bound on the number of LRs that can be supported by the system's memory and CPU availability. In large systems, naively allowing the LRs to grow to the system's available resources can lead to overfitting, which increases estimation variability. However, because GLEAM performs regularization, the LRs are merged to minimize overfitting and reduce estimation variance. Hence, GLEAM can automatically tune the number of LRs by adding and merging LRs via the seeding and regularization process.

Concept Drift Modeling: There are two components in GLEAM that impact the modeling performance of evolving streams: the kernel maintenance module KDE_{base} and the LR seeding submodule. For the KDE_{base}, the particular approach employed to handle concept drifts is intrinsic to the chosen KDE_{base} algorithm and thus is not modified by GLEAM. For example, the sample-based KDEs [Subramaniam et al. 2006; Wegman and Marchette 2003] are designed to only capture the data samples for the given time window and hence simulate the batch processing scenario as described in Section 4.4. The second component of GLEAM that impacts modeling of concept drifts is the LR seeding submodule. To accurately capture the evolving states of KDE_{base}, the internal statistics of the LRs will need to be recomputed for every state change in the KDE_{base}. Specifically, updating k elements in KDE_{base} will incur at most $O(k)$ operations on the corresponding LR to update its first and second moments. This "exact" approach can be applied for any algorithm implemented as the KDE_{base}.

This LR update cost can be reduced if the moment estimates are able to tolerate additional error. For example, the seeding module can adopt a policy to remove LRs with an average time (plus its standard deviation) older than a given threshold [Agarwal et al. 2003]. Although lower update cost is achieved, a significant amount of stale information can remain in *LR_list,* which can increase estimation errors. An improved variation would be to allow gradual fading of the LRs' weights based on the proportion of the LR's time span (defined as the average timestamp ± standard deviation) and a given minimum timestamp threshold. Similar to the prior approach, the threshold can represent the starting point of a (sliding) time window. The two variations just described can reduce the total update costs compared to the "exact" update method.

Extension for Local Region Splitting: GLEAM can be extended to support LR splitting as follows. After the seeding process, the selection of the split candidate is determined by the LR's *normalized variation*. For an LR l_j, the normalized variance is defined as $NV_j = \frac{\sigma_j}{\max(l_j)-\min(l_j)}$ where $\min(l_j)$ and $\max(l_j)$ respectively denote the minimum and maximum kernel centers in l_j. The larger the NV value, the weaker the unimodality assertion becomes. Therefore, the LR with the maximum NV can be considered for splitting. Once the split candidate l_c (i.e., LR with maximum NV) is identified, the LR is split as follows:

1. Generate all bipartite and continuous segments $l_c^{(p)}$ and $l_c^{(q)}$ where $l_c^{(p)} \cup l_c^{(q)} = l_c$, $\{l_c^{(p)}, l_c^{(q)}\} \in$ LRs, and $|l_c^{(p)}|, |l_c^{(q)}| > 1$.
2. Compute the standard deviations $\sigma_c^{(p)}$ and $\sigma_c^{(q)}$, respectively, for $l_c^{(p)}$ and $l_c^{(q)}$.
3. Find the pair of $l_c^{(p)*}$ and $l_c^{(q)*}$ where $\sigma_c - (\sigma_c^{(p)*} + \sigma_c^{(q)*}) > 0$ is maximized.
4. Replace l_c with $l_c^{(p)*}$ and $l_c^{(q)*}$ in *LR_list*.
5. Forward the *LR_list* for LR regularization.

In Step 1, the algorithm generates all pair combinations $(l_c^{(p)}, l_c^{(q)})$ of LRs such that the union of their kernel sets is equal to the kernel set of l_c. In Steps 2 and 3, the individual standard deviations of $l_c^{(p)}$ and $l_c^{(q)}$ and the reduction in the standard deviations $\sigma_c - (\sigma_c^{(p)*} + \sigma_c^{(q)*}) > 0$ are computed. This reduction in standard deviations is the logical complement of the merge criterion. Hence, the LR pairs, $l_c^{(p)*}$ and $l_c^{(q)*}$, with the maximum reduction values are selected as the new LRs. Step 4 replaces l_c with $l_c^{(p)*}$ and $l_c^{(q)*}$ in the *LR_list*. Because the split can generate new merge candidates with its neighbors, Step 5 applies regularization on the new pair and its neighbors to detect potential mergers. The total cost to find and split an LR is $O(Q_{max} + |l_c|) = O(M)$.

5.2. Query Processing with LR_query_processor

To process a density query, the *LR_query_processor* initiates the LR regularization from the *LR_modeler* to obtain the L_{reg}. Using L_{reg}, the *LR_query*_processor determines the density estimate by obtaining and aggregating the relevant sets of LRs and kernel objects. The retrieval and aggregation steps are accomplished via the following two tasks: bandwidth retrieval and kernel aggregation. In bandwidth retrieval, the

LR_query_processor retrieves the bandwidths from L_{reg} and associates them to the kernel objects in KDE_{base}. In kernel aggregation, the *LR_query_processor* combines the density contribution of the kernel objects to produce the final density estimate.

Bandwidth retrieval: For each contributing kernel object in KDE_{base}, the *LR_query_processor* must obtain the corresponding bandwidth from L_{reg}. A bandwidth query function is expressed as $BQ(x) = H_{LR}(l) \,|l \in L_{reg} \wedge (x \cap (\Omega(l) \cdots \Theta(l)] \neq \emptyset)$ where $x \in \mathcal{R}$, and $H_{LR}(l)$ is the bandwidth of LR l. An important goal of the *LR_query_processor* is to minimize the time required to search for an intersecting l (i.e., minimize $BQ(\cdot)$'s processing time). Because a bandwidth query is invoked for every contributing kernel in KDE_{base}, minimizing the cost of this step is essential to achieving good throughput performance.

Kernel aggregation: The final density estimate is achieved by aggregating the contributions of the kernel objects and its newly assigned bandwidths to the density query. The computation of the final density strictly follows the formulation of the LR-based KDE as shown in Equation (4).

To further illustrate the density estimation algorithm, an example is provided here. Assume q is a density query that is submitted to the *LR_query_processor*. The first step in generating the estimate is to produce the regularized version of *LR_list* via the *LR_modeler*. The *LR_modeler* will output L_{reg}, the regularized LR list. Second, the *LR_query_processor* will obtain all the relevant sets of kernel objects from the $KDE_{base,}$ which may include all M kernel objects. Third, for each kernel object k retrieved from the KDE_{base}, its corresponding bandwidth is obtained by invoking the bandwidth query function $BQ\left(\text{K\_center}(k)\right)$ on L_{reg}, where K_center outputs the center of k to the $\mathcal{R}$ domain. The contribution of each kernel object and its bandwidth to q are aggregated following the form as dictated by $\hat{f}_{LR}(\cdot)$ in Equation (4). Depending on the employed KDE_{base} technique, the kernel object weights will need to be considered in the calculation. In such a case, $\hat{f}_{LR}(\cdot)$ will be normalized by the aggregated kernel weights as opposed to the aggregated count $|D|$. From this example, it is clear that the query performance is highly dependent on $BQ(\cdot)$ because it is invoked for every retrieved kernel object. If a direct or linear search approach is employed for the bandwidth query, then the total density query cost is $O(M \cdot |L_{reg}|) = O(M \cdot Q_{max})$. In light of this issue, we propose a suite of optimizations (Section 5.4) that can reduce the density query cost to $O(|M - T| \cdot \log(Q_{max}))$ where $T \geq 0$.

5.3. Time and Space Cost Analyses

In this subsection, GLEAM's time and space complexities are analyzed from the perspective of its two primary tasks: LR construction and density query processing.

Local region construction: The LR construction involves two subtasks: LR seeding and LR regularization. These subtasks impact both the insertion and density query costs. Because local regularization is invoked only at query time, the insertion cost (without consideration to KDE_{base}) is solely dependent on the LR seeding. The cost of the LR seeding is equal to the combined costs of locating/inserting an intersecting LR ($O(Q_{max})$) and merging a pair of LRs ($O(Q_{max})$). Therefore, the final cost of the seeding process for each accepted data sample is $O(Q_{max})$. The total insertion cost with the KDE_{base} is $O\left(Q_{max}\right) + \text{insert\_cost}(KDE_{base})$. The insertion cost of the KDE_{base} is bounded by $O(M)$. Because $Q_{max} \ll M$ and Q_{max} are independent of M, the complete insertion cost of GLEAM with the KDE_{base} is $O(M)$.

Density query processing: For each density query, the *LR_list* is regularized, and the bandwidth of each contributing kernel object in KDE_{base} is determined. For the LR regularization phase, a copy process from *LR_list* to L_{reg} is $O\left(Q_{max}\right)$, and each scan/merge is bounded by $O(Q_{max})$. The algorithm performs at most $O\left(Q_{max}\right)$ scan and mergers. Hence, the cost of the regularization process is $O(Q_{max} + Q_{max}^2)$. The total cost of a density query is $O(Q_{max} + Q_{max}^2 + \sum_{i=1}^{M} \text{Cost}(BQ(m_i))$ where m_i is a kernel object of the KDE_{base}. If $BQ(\cdot)$ implements an exhaustive search method, the cost of calculating all the contributing kernel objects is $\sum_{i=1}^{M} \text{Cost}(BQ(m_i)) = O(M \cdot Q_{max})$. Because $Q_{max} \ll M$ and $Q_{max} \perp M$, the total cost of the density query is $O(M)$.

The space incurred by GLEAM is $O(Q_{max})$ due to the storage of *LR_list* and the generation of L_{reg} where $|L_{reg}| \leq Q_{max}$. The combined space of GLEAM is dominated by the KDE_{base}; therefore, its total space cost is $O(M)$.

5.4. Optimizations

In the following, two sets of optimization strategies are proposed to enhance the performance of the LR regularization operation and to reduce the calculation of the density contributions of kernel objects in KDE_{base}. To improve the computation of the regularization task, a heap tree is utilized to minimize the search costs for LR merge candidates. This first set of optimizations is called the *heap-based regularization*. To improve the calculation of density contributions, a second set of optimizations called *hybrid kernel aggregation and filtering* is proposed that includes techniques that perform LR buffering, LR indexing, and kernel truncation.

Heap-based Regularization Optimization: To reduce the computation of the regularization task, an optimization scheme is proposed that performs an incremental breadth-first search strategy to merge the closest candidate LR pairs. Initially, the elements of *LR_list* are copied to the LR list structure L_{reg}. Next, each pair of LRs in L_{reg} that satisfies the merge condition (i.e., $\sigma_{j\cup k} \leq \sigma_j + \sigma_k$ for nearest neighbors l_j and l_k) is inserted into a heap queue with the L_2 distance between their centers as the priority value. Now, the heap contains an initial set of merge candidate pairs from L_{reg}. Next, the lowest priority value or closest pair of LRs is dequeued from the heap. Because the dequeued object pair represents the closest LRs that satisfy $\sigma_{j\cup k} \leq \sigma_j + \sigma_k$, the pair is merged into a single LR l_m, which is performed using the vector addition operation defined previously. The merging process occurs in-situ, which immediately updates the elements in L_{reg}. If new merge candidates are formed using the combined LR l_m and its current nearest neighbors, then these candidates, along with l_m, are reinserted into the heap for potential future mergers. Otherwise, if no merge candidates exist for l_m, then the algorithm proceeds to dequeue the next pair candidates from the heap. Because the constituent LRs of l_m can appear multiple times within the heap as members of other candidate merge pairs, the algorithm must check that each dequeued pair of LRs has not been merged. If it is determined that at least one of the candidate pair has been merged, then the algorithm abandons processing of the current pair and continues to dequeue the next candidate from the heap. The regularization algorithm continues in this manner until there are no more candidates available (i.e., heap is empty). At the algorithm's termination, L_{reg} will contain the regularized LRs where $|L_{reg}| \leq |L|$ and each nearest-neighbor l_j and l_k satisfies $\sigma_{j\cup k} > \sigma_j + \sigma_k$.

The *heap-based regularization* algorithm performs at most $O(Q_{max})$ insertions/removals, and each insertion and removal from the heap is $O(\log(Q_{max}))$. Hence, the total cost of the optimized regularization is $O(Q_{max} + Q_{max}\log(Q_{max}))$, which is a significant reduction from the $O(Q_{max} + Q_{max}^2)$ cost of the direct method described in Section 5.1. Integrating this optimization strategy results in lowering the density query cost to $O(Q_{max} + Q_{max}\log(Q_{max}) + \sum_{i=1}^{M} \text{Cost}(BQ(m_i))$ where m_i is a kernel object of the KDE_{base}.

Hybrid Kernel Aggregation and Filtering Optimization: To improve the performance of the kernel density contributions, three strategies are proposed to reduce the computational requirements of the LR search and kernel aggregation. These techniques are integrated to provide the *hybrid kernel aggregation and filtering* optimization.

The first strategy, LR *buffering*, stores previously accessed LR in a cache buffer and provides direct access for nearby bandwidth queries. When a search on L_{reg} is invoked, the target (i.e., last found) LR l_t is stored in the cache buffer. As a new bandwidth query on x is invoked, l_t is checked for intersection with x, and if l_t and x intersect, then the bandwidth of l_t is returned. Here, intersection is defined to be the intersection condition of $BQ(\cdot)$ established in Section 5.2. Otherwise, if x and l_t do not intersect, then a search in L_{reg} is invoked. This buffering technique attempts to exploit bandwidth queries that are issued from neighboring kernel objects within KDE_{base}. A neighboring set of kernel objects may share a high number of LRs and therefore would benefit from the cached LR. In the best case, the buffering strategy reduces the calculation of all the contributing kernel objects to $O(M)$. In the worst case, the buffering strategy would incur $O(M \cdot Q_{max})$ time. The space cost for this technique is an additional constant for the buffer storage.

The second strategy, LR *indexing*, replaces the L_{reg} linked list with a preallocated array structure. Because the output of the LR regularization process maintains the sorted order of the regions' centroids, the LRs can be transformed to a preallocated array structure that preserves this ordering in linear time. With the preallocated array, binary searches can be performed to locate an LR within L_{reg}. Under this strategy, the cost of processing the density contribution of all kernel objects is reduced to $O(M \cdot \log(Q_{max}))$ with additional space cost of $O(Q_{max} - |L_{reg}|)$.

The third strategy, *kernel truncation*, attempts to reduce the total number of contributing kernels by pruning kernels that provide within ε error of the total density query result. A query that is sufficiently distant from a kernel k absorbs (for practical purposes) negligible contribution from k. Hence, this strategy proposes to truncate the kernel functions in order to prune the small-valued kernels. This strategy is especially useful when applied to infinitely supported kernels such the Gaussian kernel. Because the truncated kernels can introduce additional errors, a bound on these errors can be derived and parameterized via a user-defined parameter s:

$$\varepsilon \leq \frac{1}{\sum_{i=1}^{|M|} m_i} \sum_{i=1}^{|T|} w(m_i) \cdot K_{h_{min}}(sh_{min}), \tag{19}$$

where m_i is a kernel object in KDE_{base}, $w(\cdot)$ is the weight, h_{min} is the minimum bandwidth in L_{reg}, s is a threshold parameter for defining the scale of the kernel function support, and $T \leq |M|$ is the set of kernel objects for which $|x - m_i| \geq sh_{min}$.

With the truncated kernels, the *LR_query_processor* determines in $O(Q_{max})$ time the boundaries of kernel objects that will have contributions below the specified threshold (i.e., determine kernel centers with distance $|x - m_i| \geq sh_{min}$). Hence, the cost of calculating the total density contribution with this optimization is $O\,(Q_{max} \cdot |M - T|)$. No additional space is required. Combining the LR buffering, LR indexing, and kernel truncation, the cost of calculating a density contribution is $O(|M - T| \cdot \log(Q_{max}))$, which results in a total density query cost of $O(Q_{max} + Q_{max} \log(Q_{max}) + |M - T| \cdot \log(Q_{max}))$.

5.5. Multivariate Setting

To extend GLEAM to the multivariate data setting, the product kernel is considered. Equation (20) gives the form of the product kernel multivariate KDE $\hat{f}_{MKDE}$ [Scott 1992]:

$$\hat{f}_{MKDE}(\boldsymbol{x}) = \frac{1}{n} \sum_{i=1}^{n} \prod_{j=1}^{p} K_{h_j}(x_j - z_{i,j}), \tag{20}$$

where $\boldsymbol{x} = <x_1 \ldots x_p>$ is a p-dimensional query point, $z_{i,j}$ is the value of i^{th} sample in the j^{th} dimension, and h_j is the global bandwidth for the j^{th} dimension.

To extend GLEAM to the multivariate setting, an LR-based bandwidth is associated to each dimension. Let L_j be the set of LRs in dimension j; then, the multivariate LR-based KDE $\hat{f}_{MLR}$ is defined as follows:

$$\hat{f}_{MLR}(x) = \frac{1}{|D|} \sum_{i=1}^{|D|} \prod_{j=1}^{p} K_{h_{j,z_i}}(x_j - z_{i,j}), \tag{21}$$

where D is a multidimensional dataset and h_{j,z_i} is the bandwidth associated to LR $l \in L_j$, which contains $z_{i,j}$.

Because a separate set of LRs is maintained for each dimension, the approach (Section 5.1) for maintaining the LRs and its associated optimizations (i.e., heap-based regularization and LR buffering and indexing (Section 5.4)) can be applied directly. To compute a density query, the algorithms proposed for the *LR_query_processor* (Section 5.2) can be employed with some minor modifications. Specifically, for each kernel object in KDE_{base}, the intersecting LR is determined for all p dimensions and its contribution computed via the product kernel (i.e., $\prod_{j=1}^{p} K_{h_{j,z_i}}(x_j - z_{i,j})$). The kernel truncation optimization (Section 5.4) can also be utilized if the threshold s is defined for each dimension. Once the density contribution of the kernel object is attained, the process is repeated for the next object in KDE_{base}. The final density output is the summation of each kernel object contribution. Because p number of independent LR sets is maintained, modeling the LRs can be performed in parallel to improve throughput. A significant portion of the density query can also parallelized because the individual contribution can be computed independently. Hence, the multivariate LR-based KDE proposed earlier can provide an efficient solution for estimating multidimensional data streams.

Generally, data become sparser as dimensionality increases, and the sample size required to guarantee a relative MSE ≤ 0.1 at 0 is fairly small when the density is multivariate normal and the optimal bandwidth is selected [Silverman 1986]. Therefore, relatively good estimation can be attained by extracting the most representative projections from multidimensional data and then applying our proposed univariate techniques. Principal Components Analysis (PCA) is one popular technique that can be used for dimension reduction to generate projections with maximal variances [Chatfield and Collins 1990]. To apply PCA to the data stream environment, single-pass and memory constrained-based PCAs, such as those presented in Mitliagkas et al. [2013] can be utilized.

6. EXPERIMENT

Comprehensive experiments on GLEAM were conducted to evaluate the following performance elements: estimation accuracy, sample throughput, query throughput, optimization effectiveness, impact of density complexity on estimation quality, estimation under concept drift, and general applications. The experiments are organized as follows: Section 6.1 introduces the experiment design. Sections 6.2–6.4 provide an in-depth evaluation of GLEAM's effectiveness when applied to existing stream-based KDEs under the metrics of estimation accuracy, sample throughput, and query throughput. In Section 6.5, GLEAM's optimization strategies are analyzed using various baseline KDEs and datasets. In Sections 6.6 and 6.7, error analyses on the general LR-based KDE and standard KDE are performed to provide insight into the specific conditions by which the LR-based KDE can attain lower estimation error than the standard KDE. Specifically, Section 6.6 examines the relationship between the complexity of datasets, the parameter difference ratio, and the estimation quality. In Section 6.7, the estimation errors of LR-based KDE and standard KDE are compared under certain concept drift conditions. Section 6.8 illustrates general applications of GLEAM to data mining tasks such as clustering and outlier detection. Last, Section 6.9 summarizes the experimental results.

6.1. Experiment Design

This section describes the datasets, KDE algorithms and parameters, and evaluation criterion.

Datasets: To evaluate GLEAM's estimation accuracy, sample throughput, query throughput, and optimization effectiveness, three synthetic and two real-world time series datasets were employed. The synthetic datasets simulate simple to complex densities that were constructed from a mixture of normals: MIX2 $[\frac{1}{2}(N(20, 1)+ N(51, 1.35^2))]$, MIX4 $[\frac{1}{10}(N(30, 1) + N(40, 1)) + \frac{2}{5}(N(20, 1) + N(51, 1.35^2))]$, and MIX8 $[\frac{1}{8}(N(10, 1.25^2) + N(20, 1) + N(27, 2^2) + N(35, 1.4^2) + N(43, .95^2) + N(48, 3.5^2) + N(53, .75^2) + N(57, 1.5^2))]$. For each mixture, five time series instances were created. Hence, there are a total of 15 synthetic time series datasets. The samples are randomly ordered, hence the sample distributions for any two time intervals are identical. Scenarios involving time dependency are captured within the real-world data. The real-world data consist of highway loop detector measurements (HIGHWAY) [Asuncion and Newman 2007] and a power demand log from a Dutch facility (POWER) [Keogh et al. 2008]. Each time series contains 25K sample points. The true PDFs of the real-world datasets were defined as the density estimated from the AKDE using *all* available sample points.

To empirically analyze the effects of data complexity and concept drift on the LR-based KDE and standard KDE, the employed dataset must provide a broad range and high fidelity of values on the independent variables: aggregated curvature and rate of distributional change. Hence, 30 time series with different aggregated curvature values and PDFs were generated to test the effects of data complexity on estimation quality. To evaluate the estimators' performance under concept drift, 30 additional time series with evolving density structure (parameterized by the mode center distance) were synthesized. These datasets provide an extended range of values on the aggregated curvature (i.e., structural complexity) and rate of distributional change (i.e., mode center distance variation) at a finer scale.

Algorithms and Parameters: Five existing global bandwidth stream-based density estimators were evaluated including (1) Epanechnikov KDE (EKDE) [Subramaniam et al. 2006], (2) Gaussian KDE (GKDE), (3) Cluster Kernels KDE (CKKDE) [Heinz and Seeger 2008], (4) Adaptive KDE (AKDE), and (5) Histogram (HST). EKDE and GKDE are sample-based KDEs that employ the Epanechnikov and Gaussian kernel functions, respectively. As mentioned in Section 3, the CKKDE is a global bandwidth estimator that clusters/merges the sample points to maintain a finite number of kernel objects. The proposed GLEAM algorithm was applied to all the global bandwidth KDEs, and the resulting enhanced estimators are called GLEAM-EKDE, GLEAM-GKDE, and GLEAM-CKKDE. Both the optimized and nonoptimized versions of GLEAM were tested. For each global bandwidth KDE, the employed bandwidth form was Scott's Rule, and the maximum number of kernel objects was $M = 1000$. The histogram employed a bin assignment rule based on the normal reference [Scott 1992]. For the GLEAM-based KDEs, the following parameters were used: $Q_{max} = 10$, $M = 1000$, and each LR applied the Scott's Rule bandwidth. The parameter values were defined under the assumption of a resource-constrained environment that needed to generate rapid and high-quality estimates. This assumption reflects many practical applications (e.g., automatic highway incident detection) that must provide real-time analysis for multiple streams and can only provision a small amount of CPU time and memory to each stream. To simulate this environment, a relatively low M and medium Q_{max} were used. First, the M parameter was set such that a good balance of speed and space cost was achieved. Second, the Q_{max} value was set such that it did not violate the response time and space upper bounds imposed by the system.

Evaluation Criterions: The Integrated Absolute Error (IAE) is employed to quantify the estimation discrepancy. Estimation accuracy is defined as the difference between the theoretical maximum IAE and the computed IAE normalized by the theoretical maximum IAE. The following provides the formula of the estimation accuracy score:

$$A(\hat{f}) = 1 - \left(\sum_{i=0}^{1000} |\hat{f}(x_i) - f(x_i)|\Delta x\right) / E_{max}, \tag{22}$$

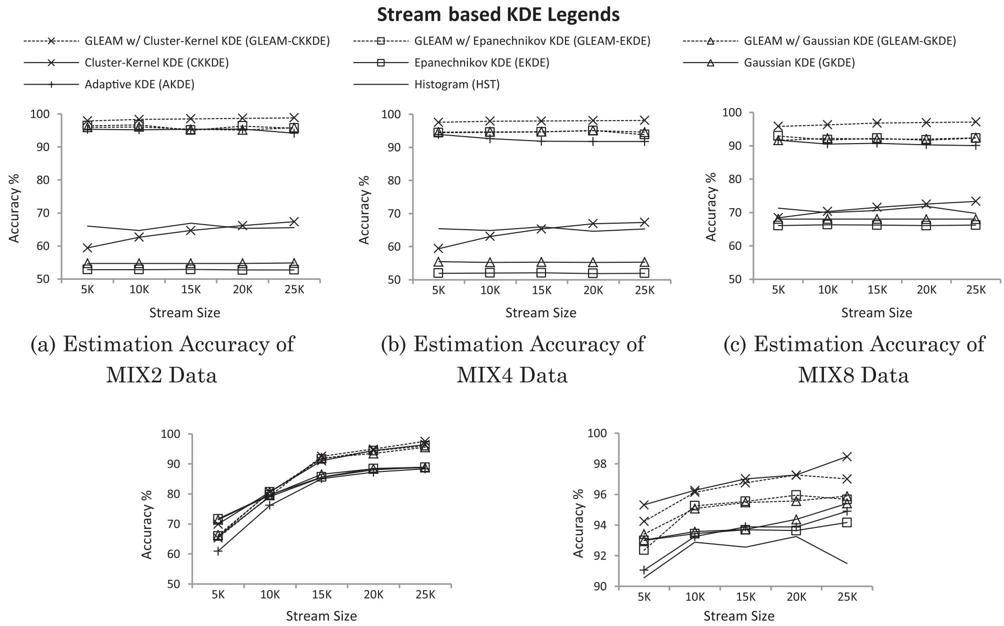
where $x_1, \ldots, x_{1000}$ are query points that uniformly divide the entire span of the distribution, $\Delta x = x_{i-1} - x_i$, and E_{max} is the theoretical maximum integrated absolute L_1 error.

The sample throughput is defined as the rate of data stream samples that can be processed for a given time unit. Likewise, the query throughput is defined as the rate of density queries that can be completed for a given time unit. The reported experiment results are the average values of these criterion measures. The calculated standard deviation % is the percentage ratio of the standard deviation and the average. The experiments were performed on a Windows Server 2003 Enterprise OS. The hardware platform was a 2.0GHz Intel Pentium Core 2 Duo with 3GB of RAM.

6.2. Estimation Accuracy

The results of the estimation accuracy are provided in Figure 3. In the graphs, the *x-axis* is the sample size, and the *y-axis* is the estimation accuracy (Equation (22)). For this set of tests, the GLEAM-based algorithms utilized the *heap-based regularization* and *hybrid kernel aggregation and filtering* optimizations. Overall, the GLEAM-based KDEs provided comparable or better (in most cases) accuracy than all the competing density estimators. The most significant gains attained by the GLEAM-based KDEs were with the MIX2, MIX4, and MIX8 datasets, which provide substantial improvements (23% to 44%) over the global bandwidth KDEs (i.e., CKKDE, EKDE, and GKDE). These large gains can be attributed to the datasets' high level of structural complexity (MIX4 and MIX8) and highly localized features (MIX2). The local bandwidth AKDE obtained similar accuracy scores under these complex datasets as the GLEAM-based estimators. However, in most cases, the GLEAM-based KDEs outperformed the AKDE. The performance of the histogram was comparable to the global bandwidth KDE techniques, but its estimation results exhibited observably higher variability than any of the KDE-based approaches.

For the POWER data, the GLEAM-based KDEs improved the accuracy of the global bandwidth estimators for data size ≥10K, with GLEAM-CKKDE achieving the highest accuracy for data size ≥15K. Observe that the estimation quality of the Cluster Kernels exceeds that of other global bandwidth KDEs. Due to the generality of the GLEAM approach, it can take advantage of such feature from the base KDE to further enhance its estimation performance. In the HIGHWAY data, the existing methods provided high-quality results with an accuracy of >90%. The GLEAM-based methods achieved comparable or improved estimates over the existing estimators. Specifically, GLEAM-GKDE and GLEAM-EKDE were able to increase the accuracies of GKDE and EKDE for data size ≥10K. For GLEAM-CKKDE, comparable scores were obtained for data sizes



(a) Estimation Accuracy of MIX2 Data

(b) Estimation Accuracy of MIX4 Data

(c) Estimation Accuracy of MIX8 Data

(d) Estimation Accuracy of POWER Data

(e) Estimation Accuracy of HIGHWAY Data

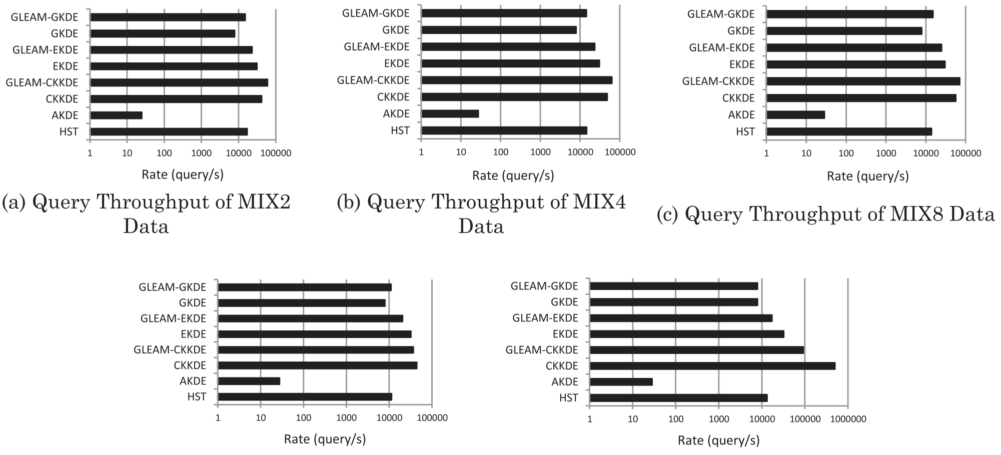
Fig. 3. Estimation accuracy of all the datasets (note the varied accuracy scale).

10K–20K. However, GLEAM-CKKDE at data sizes 5K and 25K and GLEAM-EKDE at 5K provided slightly lower scores (with $<1.5\%$ difference) than their baseline methods. This minor discrepancy indicates a potential overfitting condition for GLEAM-based methods. However, the LR regularization component was able to reduce the impact of overfitting and help to minimize the estimation error.

In summary, the GLEAM algorithm improved the accuracy in most of the datasets and exhibited the highest gains for complex densities (MIX2, MIX4, MIX8, and POWER). The GLEAM-based KDEs also inherit the advantages of the base KDE, which can further improve their estimation accuracy. The standard deviation percentage of the accuracy for all the KDE-based techniques is $\leq 2.5\%$. The standard deviation percentage of the histogram is $\leq 5\%$.

6.3. Query Throughput

Figure 4 gives the query throughput results of all the estimators. The *x-axis* is the query throughput (query/sec), and the *y-axis* is the density estimator. In most instances, the optimal GLEAM-based KDEs provided higher query throughput over their base KDEs due to GLEAM's ability to efficiently regularize the LRs and effectively prune kernel objects. Note that the GLEAM-CKKDE and CKKDE consistently achieved the highest throughputs within all the datasets. This performance advantage can be attributed to the use of a sorted index within the base KDE that allows for more aggressive pruning than the sample-based techniques. Because of its quadratic query processing cost, the AKDE exhibited the lowest query throughput (at a 10^{-3} rate of the next slowest estimator). All 25K samples were used from each dataset to measure the query throughput. The standard deviation percentage for this set of experiment is $\leq 7\%$.



(a) Query Throughput of MIX2 Data

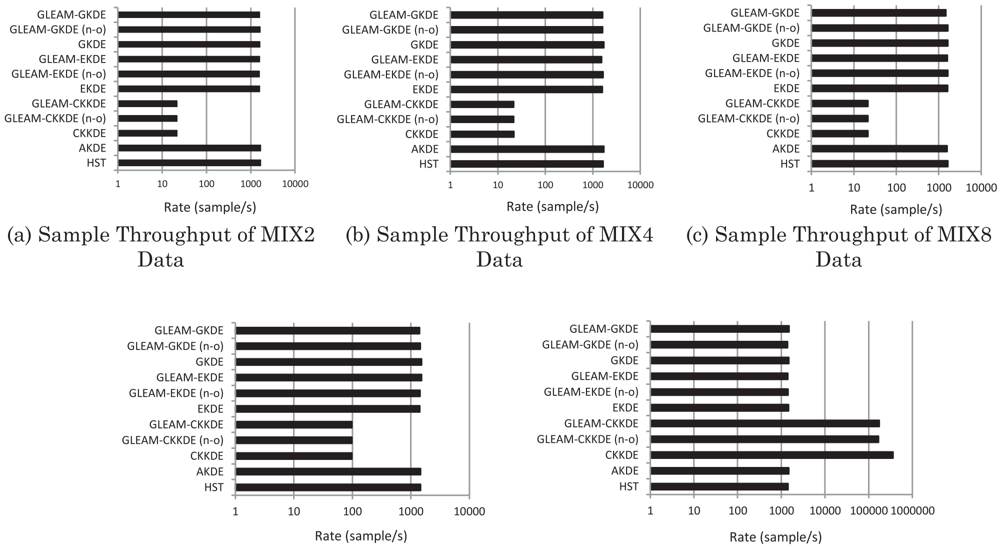
(b) Query Throughput of MIX4 Data

(c) Query Throughput of MIX8 Data

(d) Query Throughput of POWER Data

(e) Query Throughput of HIGHWAY Data

Fig. 4. Query processing performance (log scaled).



(a) Sample Throughput of MIX2 Data

(b) Sample Throughput of MIX4 Data

(c) Sample Throughput of MIX8 Data

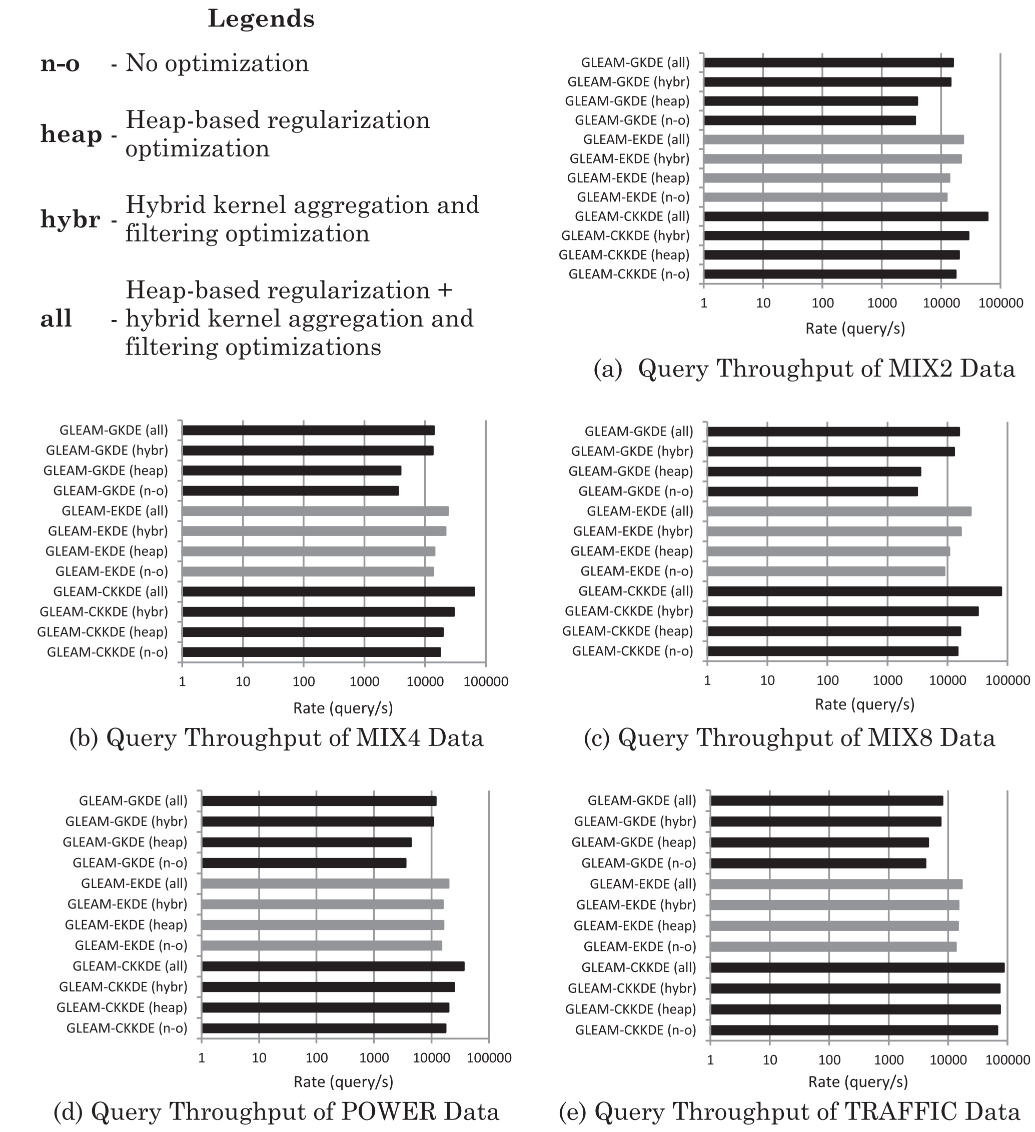
(d) Sample Throughput of POWER Data

(e) Sample Throughput of HIGHWAY Data

Fig. 5. Sample processing performance (log scaled).

6.4. Sample Throughput

The sample throughput is given in Figure 5. In the plots, the *x-axis* is the sample throughput rate (sample/sec), and the *y-axis* is the KDE technique. Here, "**n-o**" refers to the nonoptimized GLEAM-based KDEs. The GLEAM-based KDEs produced negligible overhead in the sample throughput. In fact, most of the differences between the (optimized and nonoptimized) GLEAM-based KDEs and their base KDEs are within the standard deviation. It is important to note that the cluster-based KDEs (i.e., GLEAM-CKKDE and CKKDE) attained the lowest throughput performance in all the datasets, except HIGHWAY, which achieved exceptionally high throughput. This is due to the significantly lower number of discrete values in the HIGHWAY dataset (≤100), which



(a) Query Throughput of MIX2 Data

(b) Query Throughput of MIX4 Data

(c) Query Throughput of MIX8 Data

(d) Query Throughput of POWER Data

(e) Query Throughput of TRAFFIC Data

Fig. 6. Query processing performance of various optimizations (log scaled).

effectively reduced the number kernel objects in CKKDE and GLEAM-CKKDE. In contrast, the sample-based methods provided stable throughput irrespective of the dataset. This experiment showed that GLEAM can successfully adopt the differing capabilities of the base KDEs. All 25K samples were used from each dataset to measure the sample throughput. The standard deviation percentage for this experimental component is $\leq 7\%$.

6.5. Optimization Effectiveness

This experiment tests the effectiveness of the proposed optimization strategies *heap-based regularization* and *hybrid kernel aggregation and filtering*. Figure 6 shows the query throughput of GLEAM with varying combination of optimization techniques.

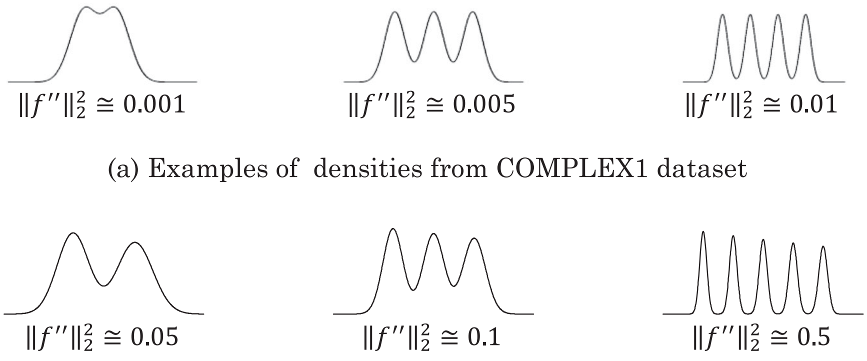
Table I. COMPLEX1, COMPLEX2, and COMPLEX3 Dataset Specification

Dataset Name	Description	# of time series	Sample size for each time series	Center c_i values for each time series	Scale s_i values for each time series	Weight w_i values for each time series
COMPLEX1	varied centers c_i, fixed scales s_i, fixed weights w_i	10	1000	$c_1 = 31$, $c_{i+1} = c_i + C \cdot i$ where $7 \leq C \leq 11$ and C unique to each time series	$s_i = S > 0\ \forall i$ where S is unique to each time series	$w_i = 1/G\ \forall i$
COMPLEX2	varied centers c_i, varied scales s_i, fixed weights w_i	10	1000	Same as COMPLEX1	$s_1 = S_0$, $s_{i+1} = s_i + S_1 \cdot i$ where $1/3 \leq S_0, S_1 \leq 13/5$ and S_0, S_1 unique to each time series	$w_i = 1/G\ \forall i$
COMPLEX3	varied centers c_i, varied scales s_i, varied weights w_i	10	1000	Same as COMPLEX1	$1/3 \leq s_i \leq 13/5$ where $s_i \neq s_j \forall i, j$	$w_1 = W_0$, $w_{i+1} = w_i + W_1 \cdot i$ where $1/10 \leq W_0, W_1 \leq 3/5$ and S_0, S_1 unique to each time series and $\sum_{i=1,\ldots,p} w_i = 1$

The *x-axis* is the query throughput (query/sec), and the *y-axis* is the GLEAM optimization technique. Within all of the datasets, the combined *heap-based regularization* and *hybrid kernel aggregation and filtering* optimizations always outperformed all other combination of optimizations. The degree of improvement generated by each optimization strategy is dependent on the employed base KDE. For example, in the TRAFFIC dataset, the *hybrid kernel aggregation and filtering* optimization provides significantly higher throughput improvement than *heap-based regularization* for GKDE but not for EKDE and CKKDE. This dramatic increase in query throughput is achieved by pruning kernel objects of the unbounded support Gaussian kernel. Overall, each optimization strategy improves the query throughput, and, when combined, the optimizations can provide significant throughput enhancement. All 25K samples were used from each dataset to measure the query throughput of different optimization strategies.

6.6. Impact of Data Complexity on Estimation Quality

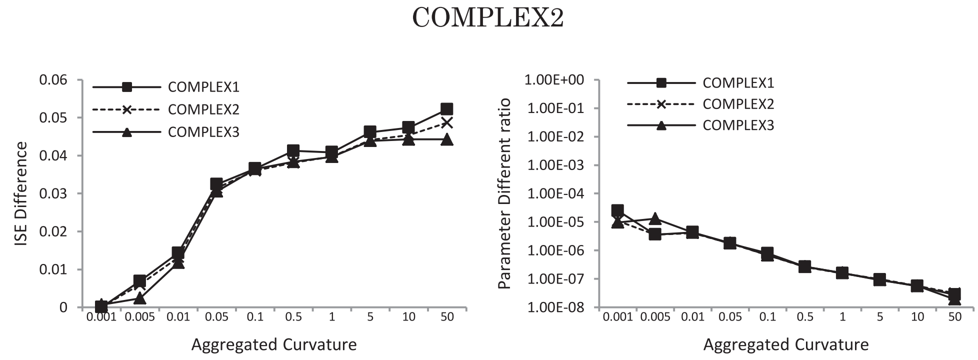
The following provides an empirical study of the relationship between the dataset's aggregated curvature $||f''||_2^2$, the AMISE disparity between the standard KDE and LR-based KDE (i.e., $Z(\hat{f}_{KDE}, \hat{f}_{LR})$), and the parameter difference ratio $\beta(K, L, D)$. In particular, this experiment intends to demonstrate the effects of data complexity (as measured by its aggregated curvature $||f''||_2^2$) on the parameter difference ratio and estimation performance of the LR-based KDE and standard KDE. Because the data complexity is the independent variable, 30 synthetic time series were generated with varying degree of aggregated curvature $||f''||_2^2$ values. The synthesized time series employed a mixture of normal density with the following canonical form: $\sum_{i=1}^{G} w_i N(c_i, s_i)$, where $G = 11$ is the maximum number of modes. Table I describes the parameters



(a) Examples of densities from COMPLEX1 dataset

(b) Examples densities from COMPLEX2 dataset

Fig. 7. Generated mixture of normals with its curvature values from COMPLEX1 and COMPLEX2.



(a) Aggregated Curvature vs. ISE Difference

(b) Aggregated Curvature vs. Parameter Difference Ratio

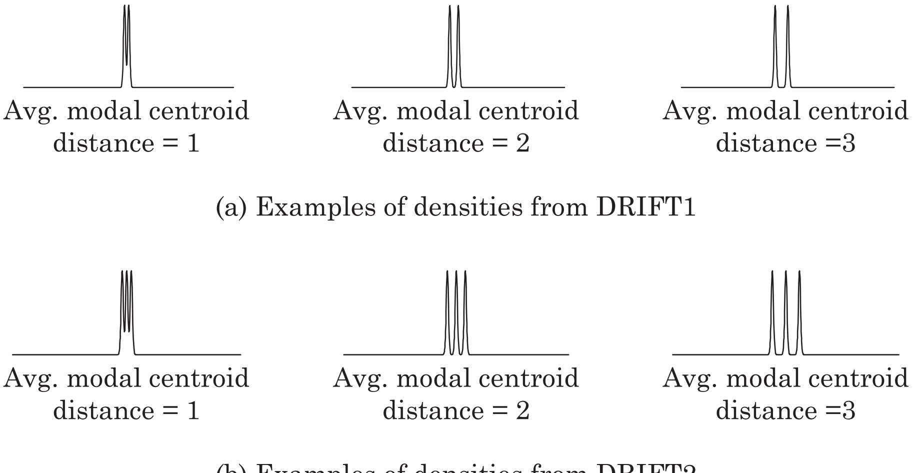
Fig. 8. Plot of the relationship between aggregated curvature ($||f''||_2^2$), ISE difference ($\approx Z(\hat{f}_{KDE}, \hat{f}_{LR})$), and parameter difference ratio ($\beta(K, L, D)$).

of the generated datasets. Figure 7 shows some examples of the densities from COMPLEX1 and COMPLEX2 datasets. For each time series, its density was estimated using the standard KDE (implemented by EKDE) and the LR-based KDE (implemented by GLEAM-EKDE). This simulation utilized the ISE measure to approximate the AMISE difference $Z(\hat{f}_{KDE}, \hat{f}_{LR})$.

Figure 8 provides the results of the simulation that demonstrate the interaction between the datasets' aggregated curvature, ISE difference, and parameter difference ratio. In each instance, when the aggregated curvature is larger than the parameter difference ratio (i.e., $||f''||_2^2 > \beta(K, L, D)$), the ISE difference between the standard KDE and LR-based KDE is positive (i.e., the LR-based KDE gave *lower* error than the standard KDE). This relationship is consistent with Theorem 4.3 and the analysis of Equation (18) in Section 4.4. Furthermore, the figure shows that when the data increase in complexity (higher $||f''||_2^2$ values), the parameter difference ratio decreases. This inverse relationship increases the disparity between the aggregated curvature and the parameter difference ratio, thus resulting in further estimation improvements

Table II. DRIFT1, DRIFT2, and DRIFT3 Dataset Specification

Dataset Name	Description	# of time series	Sample size for each time series	Center c_i values for each time series	Scale s values for each time series	Weight w values for each time series
DRIFT1	2 modes with increasing average centroid distance	10	1000	$c_1 = 40$, $c_{i+1} = c_i + C \cdot i$ where $0 \leq C \leq 9$ and C unique to each time series	$s = 1/4$.	$w = 1/2$.
DRIFT2	3 modes with increasing average centroid distance	10	1000	Same as DRIFT1	$s = 1/4$.	$w = 1/3$.
DRIFT3	4 modes with increasing average centroid distance	10	1000	Same as DRIFT1	$s = 1/4$.	$w = 1/4$.



(a) Examples of densities from DRIFT1

(b) Examples of densities from DRIFT2

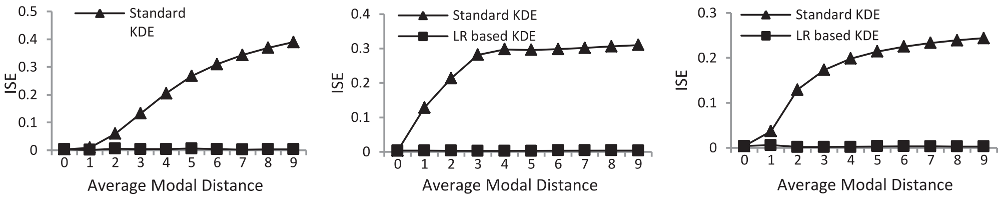
Fig. 9. Examples of densities concept drift datasets DRIFT1 and DRIFT2.

in the LR-based KDE over the standard KDE as the data's PDF becomes more complex (i.e., higher $||f''||_2^2$). From these results, the LR-based KDE shows that it can provide substantial estimation accuracy over the standard KDE for complex data streams.

6.7. Effects of Concept Drifts on Estimation Accuracy

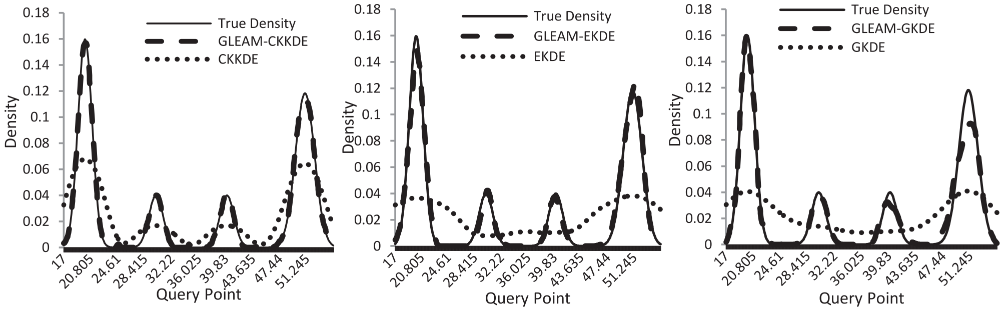
To illustrate the LR-based KDE's adaptive property, a simulation study was conducted to compare its estimation performance against the standard KDE under concept drifts. The concept drifts were generated from instances of data densities using the following form: $\sum_{i=1}^{G} w \cdot N(c_i, s)$, where G is the number of modes. The datasets have varied centers with fixed scales and weights. A detailed description of the datasets is provided in Table II. For each dataset, the average distance between the mode centers was varied at regular increments. Similar to Section 6.6, the standard KDE and LR-based KDE were implemented using EKDE and GLEAM-EKDE, respectively. Figure 9 gives some examples of the PDFs in DRIFT1 and DRIFT2 datasets.

Figure 10 shows the estimation results of the concept drift simulation. In all the datasets, the standard KDE error increases with the mode center distance, whereas



(a) Estimation quality of DRIFT1 (b) Estimation quality of DRIFT2 (c) Estimation quality of DRIFT3

Fig. 10. KDE performance under modal shifts.



(a) CKKDE, GLEAM-CKKDE estimates (b) EKDE, GLEAM-EKDE estimates (c) GKDE, GLEAM-GKDE estimates

Fig. 11. Density estimation plots of MIX4.

the LR-based KDE generally remains the same. These behaviors of the standard KDE and LR-based KDE directly support the analyses of Section 4.4. When the mode center distance is small, the LR-based KDE and standard KDE are able to attain similar performance because their error parameters are approximately equal. For example, when the mode distance is 0, the data distribution is unimodal, and hence the bandwidths of the standard KDE and LR-based KDE are the same. As the distance between the modes increases, the estimation quality of the standard KDE degrades due to the increase in σ_D. However, the LR-based KDE estimation error remains fairly constant because the increase in modal distance does not affect σ_j and $|l_j|$ of the parameter difference ratio. This observation coincides with the analytical results of Section 4.4, where the estimation error of the LR-based KDE was expected to be unaffected by the concept drift. The analyses of Section 4.4 and these simulation results show that, under modal shifts, the LR-based KDE can be employed to effectively mitigate the effects of changing density structures.

6.8. General Applications

This subsection discusses some general applications of GLEAM to popular data mining tasks, including outlier detection and clustering within the data stream environment. Figure 11 provides the density plot of the MIX4 dataset and its estimates from EKDE, GKDE, CKKDE, and their GLEAM versions for all 25K samples. In all cases, the GLEAM estimators provide highly accurate models (accuracy >90%) and drastically improve on the baseline KDEs. For examples, GKDE fails to correctly estimate the two modes in the center, whereas EKDE gives a misleading model by estimating a single mode between the true modes. CKKDE is able to capture the structure of all the modes but at much lower throughputs than the GLEAM version, GLEAM-CKKDE (see

Sections 6.3 and 6.4). Applying these density estimates to clustering (via a mode finding method) would result in potentially significant false dismissals. In some surveillance scenarios, clustering is employed as a mechanism to represent the baseline behavior of the environment sensed from commodity and single-mode (i.e., one dimensional) sensors. The dismissed clusters would lead to reduced sensitivity to true changes in the environment and lower surveillance effectiveness.

In a related context, distance-based outliers can be estimated by utilizing the probability of a given sample interval falling below a prespecified threshold [Subramaniam et al. 2006]. Now consider the case of some of the sample points falling within the troughs of the true density and regard these points as the true outliers. If a stream-based estimator employs one of the existing KDE methods, EKDE, GKDE, or CKKDE, then these outlying samples would have a greater chance of being misclassified as normal samples than an estimator using GLEAM because the density values of these sample points would be overestimated by the existing techniques. Because GLEAM is able to model the troughs of the density with much higher accuracy ($\geq$25%) than existing methods, GLEAM can produce significantly lower misclassification rate. Hence, within these usage scenarios, it can be seen that GLEAM's superiority in estimation quality over existing stream-based KDEs can lead to nontrivial improvements in the effectiveness of the data mining applications.

6.9. Discussion

The experiments in Section 6.2 demonstrated that applying GLEAM to the existing set of stream-based KDEs can dramatically improve the estimation quality of structurally complex distributions (up to 44%). In addition, through its LR regularization, GLEAM has been shown to effectively minimize the overfitting problem by providing comparable or better estimation accuracy for simple densities (e.g., HIGHWAY). Because data streams have a high propensity to mutate, it is critical to the overlying mining operations that the density estimators accurately capture the consequent changes in their density structure. For mining tasks such as concept drift detection, where its detection performance is crucially dependent on the underlying estimator's (modeling) capacity, the appearance of unseen local features can cause detection failure if the estimator is unable to appropriately model a variety of complex distributions.

GLEAM's ability to improve modeling accuracy is complemented by its efficient approach to sample and query processing. As shown in Sections 6.3–6.5, GLEAM, with its *heap-based regularization* and *hybrid kernel aggregation and filtering* optimizations, can adopt and improve the throughput of the base KDEs while bounding it to $O(M)$ asymptotic worst-case performance. An example of GLEAM's ability to retain the critical features of its base KDE is seen in the HIGHWAY dataset. In this dataset, GLEAM-CKKDE outperformed all the noncluster-based KDEs in query throughput, whereas GLEAM-EKDE and GLEAM-GKDE retained the sample throughput consistency of the sample-based approaches. This property of GLEAM provides for a high level of versatility that is not present in any existing stream-based KDEs.

The simulation study in Section 6.6 demonstrated that the LR-based KDE provided lower estimation error than did the standard KDE as the dataset complexity increases (i.e., aggregated curvature value increases). Furthermore, the condition for which this improvement occurs is made to be precise by analyzing the relationship between the data's aggregated curvature and parameter difference ratio.

The empirical analysis of Section 6.7 investigated the performance of the LR-based KDE under a common data stream scenario: concept drifts. Mode center variation can occur in practice, such as in highway traffic. For example, some roadway constructions shift the mode centers of normal traffic patterns. The experiments showed that the

LR-based KDE, as implemented by GLEAM, can effectively adapt to these structural mutations to produce stable estimation.

7. CONCLUSION

This article provides in-depth analyses of the LR-based KDE to derive important properties such as the AMISE, conditions for attaining lower estimation error than the standard KDE, and application to existing stream-based KDEs. Based on the analyses, the generalized LR-based algorithm (GLEAM) framework is proposed that can be applied to existing stream-based KDEs to enhance the estimation accuracy of structurally complex distributions. The bias-reducing qualities of the LR-based approach are justified through analyses of its estimation errors. Consistent with the theoretical analyses, experiments have shown that the GLEAM framework effectively improved the estimation quality of existing techniques under a variety of datasets and especially those that possess complex distributions. Optimization strategies are proposed to reduce the query processing overhead, which results in consistent throughput improvements over the standard version. Furthermore, GLEAM combined with the proposed optimizations is guaranteed to process any density query in at most linear time. Due to the generic nature of GLEAM, it can effectively leverage the characteristics of its base KDE to provide an unprecedented level of versatility to support a wide array of stream mining applications.

REFERENCES

C. Aggarwal. 2003. A framework for diagnosing changes in evolving data streams. In *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data*. 575–586.

C. Aggarwal, J. Han, J. Wang, and P. S. Yu. 2003. A framework for clustering evolving data streams. In *Proceedings of the 2003 International Conference. on Very Large Data Bases (VLDB'03)*.

C. Aggarwal and P. S. Yu. 2007. A survey of synopsis construction in data streams. In *Data Streams: Models and Algorithms*, C. Aggarwal, Ed. Springer Science and Business Media, New York, 169–202.

A. Asuncion and D. J. Newman. 2007. UCI machine learning repository. University of California, School of Information and Computer Science, Irvine, CA. Available at http://www.ics.uci.edu/∼mlearn/MLRepository.html.

B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom. 2002. Models and issues in data stream systems. In *Proceedings of the 21st ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems.* 1–16.

A. P. Boedihardjo, C. T. Lu, and F. Chen. 2008. A framework for estimating complex probability structures in data streams. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management (ACM CIKM)*. 619–628.

A. Bowman. 1984. An alternative method for cross-validation for the smoothing of density estimates. *Biometrika* 71, 353–360.

C. Chatfield and A. J. Collins. 1990. *Introduction to Multivariate Analysis*. Chapman & Hall.

P. Domingos and G. Hulten. 2012. A general framework for mining massive data streams. *Journal of Computational and Graphical Statistics* 12, 945–949.

M. Garofalakis, J. Gehrke, and R. Rastogi. 2002. Querying and mining data streams: You only get one look (tutorial). In *Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data.* 635–635.

P. Gibbons, Y. Matias, and V. Poosala. 2002. Fast incremental maintenance of approximate histograms. *ACM Transactions on Database Systems* 27, 261–298.

A. Gilbert, Y. Kotidis, S. Muthukrishan, and M. J. Strauss. 2002. How to summarize the universe: Dynamic maintenance of quantiles. In *Proceedings of the 28th International Conference of Very Large Data Bases.* 454–465.

A. Gray and A. Moore. 2003. Rapid evaluation of multiple density models. In *Proceedings of the 9th International Workshop on Artificial Intelligence and Statistics.*

S. Guha, N. Koudas, and K. Shim. 2006. Approximation and streaming algorithms for histogram construction problems. *ACM Transactions on Database Systems* 31, 396–438.

P. Hall, S. N. Lahiri, and Y. K. Truong. 1995. On bandwidth choice for density estimation with dependent data. *Annals of Statistics* 23, 2241–2263.

P. Hall and J. S. Marron. 1987. Estimation of integrated squared density derivatives. *Statistics and Probability Letters* 6, 109–115.

P. Hall, S. J. Sheather, M. C. Jones, and J. S. Marron. 1991. On optimal data-based bandwidth selection in kernel density estimation. *Biometrika* 78, 263–269.

W. Hardle, M. Muller, S. Sperlich, and A. Werwatz. 2004. *Nonparametric and Semiparametric Models*. Springer-Verlag, Germany.

T. Hastie, R. Tibshirani, and J. Friedman. 2001. *The Elements of Statistical Learning*. Springer-Verlag.

N.-B. Heidenreich, A. Schindler, and S. Sperlich. 2010. Bandwidth selection methods for kernel density estimation: A review of performance. In *Social Science Research Network, Social Science Electronic Publishing.* 1–28.

C. Heinz and B. Seeger. 2006. Towards kernel density estimation over streaming data. In *Proceedings of the 13th International Conference on Management of Data.* 91–102.

C. Heinz and B. Seeger. 2008. Cluster kernels: Resource-aware kernel density estimators over streaming data. *IEEE Transactions on Knowledge and Data Engineering* 20, 880–893.

N. L. Hjort and M. C. Jones. 1996. Locally parametric nonparametric density estimation. *Annals of Statistics* 24, 1619–1647.

Y. Ioannidis. 2003. The history of histograms (abridged). In *Proceedings of the 29th International Conference on Very Large Databases.* 19–30.

M. C. Jones, J. S. Marron, and S. J. Sheather. 1996. A brief survey of bandwidth selection for density estimation. *Journal of the American Statistical Association* 91, 401–407.

E. Keogh, X. Xi, L. Wei, and C. A. Ratanamahatana. 2008. The UCR time series classification/clustering. Available at http://www.cs.ucr.edu/∼eamonn/time_series_data.

E. M. Knorr and R. T. Ng. 1998. Algorithms for mining distance-based outliers in large datasets. In *Proceedings of the 24th Very Large Databases Conference*, New York, 392–403.

E. Lehmann. 1998. *Theory of Point Estimation*. Springer, New York.

C. R. Loader. 1996. Local likelihood density estimation. *The Annals of Statistics* 24, 1602–1618.

C. R. Loader. 1999. Bandwidth selection: Classical or plug-in? *Annals of Statistics* 27, 415–438.

I. Mitliagkas, C. Caramanis, and P. Jain. 2013. *Streaming, Memory-limited PCA.* University of Texas at Austin.

L. O'Callaghan, N. Mishra, A. Meyerson, S. Guha, and R. Motwani. 2002. Streaming-data algorithms for high-quality clustering. In *Proceedings of the 18th IEEE International Conference on Data Engineering.* 685–694.

A. Okabe, T. Satoh, and K. Sugihara. 2009. A kernel density estimation method for networks, its computational method and GIS-based tool. *International Journal of Geographical Information Science* 23, 1–31.

E. Parzen. 1962. On estimation of a probability density function and mode. *Annals of Mathematical Statistics* 33, 1065–1076.

M. Rudemo. 1982. Emperical choise of histograms and kernel density estimation. *Scandanavian Journal of Statistics* 9, 65–78.

S. Sain. 1994. Adaptive kernel density estimation. In *Statistics.* Rice University, Houston.

D. W. Scott. 1992. *Multivariate Density Estimation*. Wiley & Sons, New York.

S. J. Sheather and M. C. Jones. 1991. A reliable data-based bandwidth selection method for kernel density estimation. *Journal of the Royal Statistical Society* 53, 683–690.

B. W. Silverman. 1986. *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, London.

C. J. Stone. 1984. An asymptotically optimal window selection rule for kernel density estimates. *Annals of Statistics* 12, 1285–1297.

S. Subramaniam, T. Palpanas, D. Papadopoulos, V. Kalogeraki, and D. Gunopulos. 2006. Online outlier detection in sensor data using non-parametric models. In *Proceedings of the 32nd International Conference on Very Large Databases.* 187–198.

B. A. Turlach. 1993. Bandwidth selection in kernel density estimation: A review. C.O.R.E. and Institut de Statistique, Universite Catholique de Louvain, Louvain-la-Neuve, Belgium.

P. Van Kerm. 2003. Adaptive kernel density estimation. *Stata Journal* 3, 148–156.

E. J. Wegman and D. J. Marchette. 2003. On some techniques for streaming data: A case study of internet packet headers. *Journal of Computational and Graphical Statistics* 12, 1–22.

Z. Xie and J. Yan. 2008. A kernel density estimation of traffic accidents in a network space. *Computers, Environment, and Urban Systems* 35, 396–406.

T. Zhang, R. Ramakrishnan, and M. Livny. 1996. BIRCH: An efficient data clustering method for very large databases. In *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*. 103–114.

T. Zhang, R. Ramakrishnan, and M. Livny. 1999. Fast density estimation using CF-kernel for very large databases. In *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 312–316.

A. Zhou, Z. Cai, L. Wei, and W. Qian. 2003. M-Kernel merging: Towards density estimation over data streams. In *Proceedings of the 8th International Conference on Database Systems for Advanced Applications*. 285–292.

Received June 2012; revised June 2013; accepted January 2014