Efficient Algorithms for Mining Data Streams

Arnold P. Boedihardjo

Dissertation submitted to the faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Computer Science and Applications

Chang-Tien Lu, Chair
Ing-Ray Chen
Weiguo Fan
Yao Liang
Naren Ramakrishnan

August 10, 2010
Falls Church, Virginia, USA

Keywords: Data Stream, Data Mining, Machine Learning, Kernel Density Estimation, Outlier Detection

Efficient Algorithms for Mining Data Streams

Arnold P. Boedihardjo

Abstract

Data streams are ordered sets of values that are fast, continuous, mutable, and potentially unbounded. Examples of data streams include the pervasive time series which span domains such as finance, medicine, and transportation. Mining data streams require approaches that are efficient, adaptive, and scalable. For several stream mining tasks, knowledge of the data's probability density function (PDF) is essential to deriving usable results. Providing an accurate model for the PDF benefits a variety of stream mining applications and its successful development can have far-reaching impact to the general discipline of stream analysis. Therefore, this research focuses on the *construction of efficient and effective approaches for estimating the PDF of data streams*.

In this work, kernel density estimators (KDEs) are developed that satisfy the stringent computational stipulations of data streams, model unknown and dynamic distributions, and enhance the estimation quality of complex structures. Contributions of this work include: (1) theoretical development of the local region based KDE; (2) construction of a local region based estimation algorithm; (3) design of a generalized local region approach that can be applied to any global bandwidth KDE to enhance estimation accuracy; and (4) application extension of the local region based KDE to multi-scale outlier detection. Theoretical development includes the formulation of the local region concept to effectively approximate the computationally intensive adaptive KDE. This work also analyzes key theoretical properties of the local region based approach which include (amongst others) its expected performance, an alternative local region construction criterion, and its robustness under evolving distributions. Algorithmic design includes the development of a specific estimation technique that reduces the time/space complexities of the adaptive KDE. In order to accelerate mining tasks such as outlier detection, an integrated set of optimizations are proposed for estimating multiple density queries. Additionally, the local region concept is extended to an efficient algorithmic framework which can be applied to any global bandwidth KDEs. The combined solution can significantly improve estimation accuracy while retaining overall linear time/space costs. As an application extension, an outlier detection framework is designed which can effectively detect outliers within multiple data scale representations.

ACKNOWLEDGEMENTS

I would like to give my gratitude to my advisor, Dr. Chang-Tien Lu, whose tremendous generosity and guidance has made this dissertation possible. His intimate knowledge of computer science and vast familiarity in related fields have helped shape the direction and course of this research work. His continuous encouragement, support, and patience have given me the sustenance to persevere through the (at times, seemingly insurmountable) challenges of this PhD program. I could not have hoped for a better advisor.

I would also like to thank my committee members, Dr. Chen, Dr. Fan, Dr. Liang, and Dr. Ramakrishnan for their constructive criticisms and guidance. Through their involvement, they have extended both the thoroughness and applicability of this research work. Their insightful perspectives have helped seed some exciting venues for future research.

I have also gained many wonderful friends in the course of my study, Feng Chen, David Dai, Yufeng Kou, and Manu Shukla. They have helped me to better grasp my problem domain and to foster new solutions.

Lastly, I would like to thank my parents, sister, and my girlfriend, Allison for their unconditional support throughout my graduate study. They have witnessed my worst but rarely my best, and yet never ceased to provide me encouragement. Without them, none of this would have been possible.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1. INTRODUCTION

The progress in hardware and software technology has allowed agencies to deploy sensor networks, store large databases, and implement extensive information pipelines. These technical advancements represent some of the key motivating elements in the need to develop large-scale and robust data management solutions. A vital component in managing these massive amounts of information is the integration of tools which can process data rapidly and efficiently. Processing large volumes of information is a challenging task and in many instances, the processing must be done *online* and results furnished in *real-time*. For the online scenario, data arrive as *streams* which are fast, continuous, mutable, ordered, and potentially unbounded [11]. Several instances of data streams stem from the ubiquitous time series which span domains such as finance, medicine, and transportation. Some well-known streams are traded stock prices, measures of brain electrical impulses (e.g., electroencephalograms), and roadway performance metrics (e.g., vehicle speed). However, the correspondence of streams to real-time events or their adherence to an intrinsic temporal ordering (e.g., time series) does not constitute the necessary condition for its existence. For example, data streams can be constructed from the continuously scanned outputs of a database. Hence, it can be observed that several problems within the offline setting can be casted into data stream problems. Due to the large and fast nature of data streams, developing efficient techniques to process and manage this information has become a common and prevailing goal of stream research.

In order to study and analyze a particular stream phenomenon, a common approach is to choose a set of mining concepts and apply them to the data of interest. In general, stream mining attempts to elicit some specific and useful knowledge of the data, a goal that is shared by the traditional offline mining approaches. However, stream mining further subsumes the general data stream policy for rapid processing and efficient data management. Because most stream applications operate under real-time environments, this policy helps to ensure that mined results are delivered promptly and efficiently. For example, in the context of transportation data, a stream-based outlier detection algorithm is used to detect time critical traffic incidents [18]. In a related work, concept shifts are continuously tracked from road sensors to warn highway operators of pre-incident conditions [65]. Hence, it can be seen that stream mining plays an important role in enhancing the use and impact of data stream applications.

1.1 Motivation

It is well known that the structure of a data distribution is completely expressed by its *probability density function (PDF)* [39, 82]. Hence, knowledge of the PDF is an essential precursor to many data mining algorithms. For instance, in pattern classification, generative learning approaches employ the PDF to predict a sample's class by selecting the class with maximum posterior probability. In addition, the posterior probability is obtained from the prior and conditional probabilities which are described by the joint PDF [40]. Likewise, for several stream mining approaches, knowledge of the data's PDF is the primal requirement for their effective implementations. Some concrete examples of real-time stream

mining applications which employ the PDF include: outlier detection by modeling a sensor's sample distribution and selecting points with low probability values [86]; concept drift detection via comparison of current and past data streams' PDF estimates [5]; and pattern discovery in Internet traffic by visualizing the estimated PDF of arriving packets [91]. In summary, the probability density function serves as a crucial element to a wide range of real-time stream mining tasks. The challenge then is to seek a systematic method by which one can obtain a data stream's PDF that optimizes both the estimation quality (i.e., accuracy) and cost efficiency. As a further challenge, it is a rarity in data streams that the form of the density function is known. If the form is given at a particular snapshot of time, it is difficult to infer the structure for any other time due to the stream's potentially evolving nature. Employing an incorrect probability density model can result in severe accuracy loss and render the mined outcome useless.

1.2 Problem Statement

This research aims to investigate and develop efficient and effective PDF estimation techniques for data streams. Because most data streams are placed within real-time settings, this work focuses on the study and development of in-memory algorithms. Consequently, this assumed platform implies specific restrictions on the available storage and computational resources. Hence, the efficiency of stream-based approaches will be evaluated on the fulfillment on its set of space and computational constraints under the worst case scenarios. Lastly, the algorithms' effectiveness will be evaluated on its accuracy as specified by the stream application.

To frame the problem statement into context, the following definitions and assumptions are provided.

Symbol	Description
D	A set of ordered independent and identically distributed data samples
$\tau(\cdot)$	Time stamp function of a given data sample
$f(\cdot)$	Probability density function of a D
$\hat{f}(\cdot)$	Estimator of $f(\cdot)$
M	The size of the synopsis structure utilized by estimator $\hat{f}(\cdot)$ to describe D

Table 1.1 Notations and descriptions

A formal description of the density estimation problem is given as follows: *Let any given data stream be represented by the independent and identically distributed (i.i.d.) sample set* $D = \{x^{(1)}, x^{(i)}, .., x^{(n)}\}$ *where* $x^{(i)}$ *is a real-valued scalar and each* $x^{(i)}$ *is associated with a timestamp* $\tau(x^{(i)}) \in [\tau_{min}, \tau_{max}]$. *Furthermore, assume that an unknown density function* $f(\cdot)$ *exists which generates the elements (i.e., realizations) of* D. *The task is to construct and maintain an estimator* $f(\cdot)$ *with storage,*

update, and query costs that are bounded by $O(M)$ *where* $M \ll \|D\| = n$.

1.3 Research Issues

The properties of data streams highlight a seeming contradiction to the undertaken objective of this research. The following data stream characteristics comprise of the key challenges of this research [5, 7, 11]:

- Potentially unbounded stream size
- Rapid data arrival rate
- Generally unknown data distribution form
- Evolving data distribution

Traditional probability density estimation techniques cannot be directly applied to the data stream environment due to the following reasons: Firstly, classical approaches assume that processing is done offline and demand large amounts of memory that can require time costly access to secondary storage. Secondly, the primary aim of most offline methods is to maximize accuracy which often leads to algorithms that exhibit high computational costs. Thirdly, the traditional techniques assume that the dataset remains static in both cardinality and distribution which eliminates the possibility of evolving structures. The deficiencies of traditional techniques further highlight the need for developing stream-based algorithms.

There are three major issues that this research addresses in developing efficient and effective stream-based probability density estimates. These major research issues are as follows:

1. **Time and space complexity –** Heed the worst case time and space complexities permitted by the stream application.
2. **Accuracy and performance parameterization –** Generate accurate density estimates under the given time/space constraints and design capabilities for balancing between algorithmic complexity and estimation quality.
3. **Unknown structure and data evolution –** Develop methods to estimate structures not known a priori and formulate approaches to capture evolving structures.

1.3.1 Time and Space Complexities

Real-time data stream applications are constrained to limited storage and computational resources. Specifically, the employed algorithms are subject to the constraints of *fixed memory space* and *linear order scans of the dataset* [11, 31, 64]. The first restriction is a natural consequence of the in-memory nature of real-time stream algorithms. Due to a variety of reasons, the amount of available in-memory space is usually small (e.g., multiple tasks on a central stream server or sensor node level processing). A

secondary storage may be employed within the stream system, however its use can only serve as an auxiliary component to its real-time operations (e.g., logging). Therefore, the use of a secondary storage should not degrade the run-time performance of its critical functions. Due to the small and fixed-size memory, an entire data stream cannot be fitted into main memory. This restriction results in only allowing a linear order scan of the arriving stream. The stream algorithm should also provide asymptotic time guarantees on its set of operations. This constraint may vary from one application to another, but in the ideal case, the order of any single operation should be at most linear to the stream size.

1.3.2 Accuracy and Performance Parameterization

The limitations on space and run-time provide a trade-off to the accuracy of estimation algorithms. This trade-off is highly apparent in the histogram-based density estimator. The histogram can yield better accuracy if both the data samples and the number of bins are increased, however, increasing the number bins also raises the time complexity. Hence, an important research issue is to investigate and construct compact but accurate representations of the data. Furthermore, as more information are compressed (e.g., sufficient statistics for various model forms), the operations required to extract the solutions (i.e., estimates) are potentially more sophisticated which may result in higher computational costs. To balance this trade-off, the algorithms should explicitly support functions that will allow some tuning between accuracy and time/space costs.

1.3.3 Unknown Structure and Data Evolution

There are various instances in data streams where the probabilistic structures (i.e., underlying density forms) are unknown. For example, the rate of page hits of a newly deployed site or a patient's heart rate after receiving a new treatment. Structures in data streams can vary in complexity: from distributions that are unimodal to those that are multimodal and multi-scaled. Under these scenarios, the estimation technique may employ a nonparametric approach to model the underlying density. However, most of the nonparametric density estimation techniques are designed to work within an offline setting and therefore do not give emphasis on the economical use of space and computational resources. Furthermore, the approaches cannot directly describe evolving structures because the techniques assume that the data is static. This work aims to investigate methods that can effectively synopsize the data and reduce the necessary computations for generating estimation results. The existence of evolving streams will require design and analysis of strategies that can efficiently update the nonparametric model. Update to the model may involve re-computation of the smoothing parameter which has tremendous impact on the runtime performance and accuracy of the estimator.

1.4 Contributions

This research proposes the following contributions in the area of data stream mining:

A. **Theoretical development of the local region concept for kernel density estimation:** Propose the concept of local regions to effectively approximate the smoothing parameter of the nonparametric adaptive kernel density estimator. The approximation scheme will support the development of algorithms that can reduce the time complexity of a density query from $O(n^2)$ to $O(n)$. Derive fundamental theoretical properties of the local region based approach. The results are used to evaluate the estimator's expected performance, provide the precise the conditions by which enhancements in estimation accuracy can be obtained, and propose an alternative local region construction metric. Application of the local region concept to kernel estimators establishes a set of nonparametric estimation techniques which can be applied to evolving data streams with arbitrary structures, provide linear order computation, and allow parameterization of the estimator's accuracy/performance.

B. **Design of the Local Region Kernel Density Estimator (LR-KDE):** Propose a local region based density estimation algorithm (LR-KDE) which can efficiently model data streams under a sliding window. Empirical results demonstrate that the proposed algorithm provides higher estimation quality than existing techniques. Additionally, analysis shows that the proposed technique is asymptotically consistent, which until now, has been primarily attributed to the stream-based and global bandwidth kernel density estimators. The LR-KDE aggregates the results of multiple local nonparametric estimators and thus can effectively model any arbitrary distribution. Furthermore, the LR-KDE employs a local region management strategy that attains a linear order worst-case query cost. Lastly, the accuracy and runtime performance of the LR-KDE can be adjusted via the local region number.

C. **Development of the multiple density query framework for local region based KDE:** Design a framework to efficiently process multiple density queries within the local region based KDE. This framework will enhance the efficiency of stream mining tasks such as outlier detection by reducing the response times required to estimate multiple densities. To reduce computation, the framework employs an interpolative technique to generate several estimates at a time. An exact multiple density query algorithm is proposed to efficiently generate the control points for interpolation. Because the control points are critical to maximizing the accuracy of the estimates, a number of strategies are developed which can accurately model a variety of query distributions. The strategies include uniform gridding of queries, query sub-sampling, and local uniform gridding.

D. **Construction of the Generalized Local rEgion AlgorithM (GLEAM):** Applying the theoretical results of the local region concept, a generalized and efficient algorithmic framework (GLEAM) is proposed to enhance the accuracy of existing stream-based estimators. GLEAM is the first generalized approach that can be applied to existing stream-based to reduce estimation error of complex distributions while achieving linear order runtime and space efficiency. Because GLEAM employs the local region estimation concept, it can effectively model any arbitrarily shaped distribution. In contrast to the LR-KDE, a regularization constraint on the local region number is constructed to minimize data over fitting.

E. **Application extensions of the local region based kernel density estimator:** Using statistics of

the probability density function, a traffic incident detection system is developed which can operate within an evolving stream environment, requires small training dataset, and efficiently uses the system's computational resources. Additionally, the local region based kernel density estimator is applied to the problem of online multi-scale outlier detection. Because outlier detection operations must process multiple query sets, a new algorithmic approach is developed to efficiently resolve these queries within the local region based kernel density estimator.

1.5 Thesis Organization

The remainder of this thesis is organized as follows. Chapter 2 provides the background on data stream mining. Chapter 3 describes the proposed stream-based nonparametric local region based kernel density estimator. Chapter 4 provides in-depth theoretical analysis of the local region based kernel density estimators. Based on the theoretical findings, Chapter 5 introduces the GLEAM approach which can be applied to existing stream-based estimators. Chapter 6 provides a discussion on applications of the proposed kernel density estimators to address the problem of outlier detection. Chapter 7 summarizes the research achievements, publications, and future work.

Chapter 2. FUNDAMENTALS OF STREAM MINING

This chapter describes the foundational concepts of data stream mining. The organization of this chapter is as follows: Section 2.1 elucidates the requirements and constraints of the data stream mining system (DSMS) framework. Sections 2.2 and 2.3 provide details on the major components of the DSMS framework: *stream processing* and *mining* modules. A review of existing DSMS applications is given in Section 2.4. Section 2.5 provides the connection between DSMS and density estimation and elaborates on its design and operational requirements.

2.1 Data Stream Mining System (DSMS) Framework

Data stream mining is the process of discovering novel patterns, rules, and trends within an arriving and ordered set of data that is fast, continuous, mutable, and potentially unbounded [11]. A data stream can possess multiple dimensions which include common attributes such as time and location. Due to the inbound nature of data streams, an implicit time order can be imposed on the dataset. This property of data streams has allowed numerous applications on the ubiquitous time series data. Some examples of data streams are the locations of a continuously moving vehicle, observed temperature readings of a sensor node, and prices of traded financial stocks. The goals of the DSMS are a combination of the directives of traditional data mining and general stream processing [11, 31, 64]. Hence, the aims of the DSMS are as follows:

1. **Fast processing of fundamental queries –** enable efficient processing of primitive queries on the data stream necessary for the mining task. An example is the join query, where joined results between multiple data streams are used as precursors to perform a particular mining task.
2. **Efficient mining of patterns, rules, or trends –** provide a timely and compact procedure to compute results of the defined mining task.

Subject to the following constraints

- Small memory space
- Limited computational power
- Finite size time window

In order to generate solutions rapidly under the restrictions of limited memory and computational resources, the resultant operation of the DSMS should possess at most an asymptotic time complexity that is linear to the data size. This requirement has caused existing works to shift its focus from rendering exact and computationally intensive solutions to employing approximating strategies. The involved components in generating approximate solutions are illustrated in the DSMS framework (Figure 2.1).

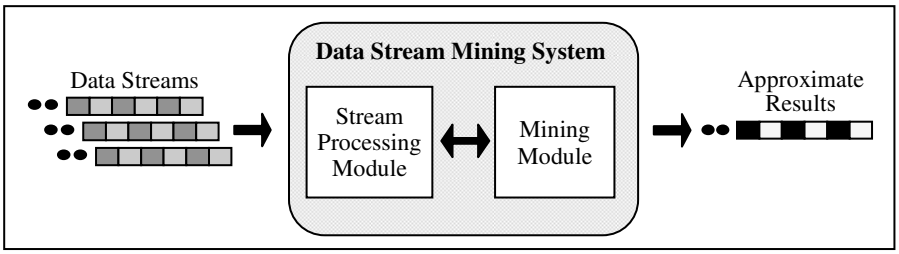


Figure 2.1 Data Stream Mining System Framework

In the DSMS framework, incoming data stream (single or multiple streams) arrive at the *stream processing module*. Summarized statistics and direct access to subset data stream (e.g., stored in the form of a sliding window) are computed and made available to the remaining subsystems. The *mining module* employs these sets of information to perform a specific mining task (e.g., classification). The following sections elucidate the key concepts of the *stream processing module* and *mining module*.

2.2 Stream Processing Module

This component (Figure 2.2) is responsible for the *data preparation*, *data selection*, and *data summarizations*. In the following, these components are described in detail.

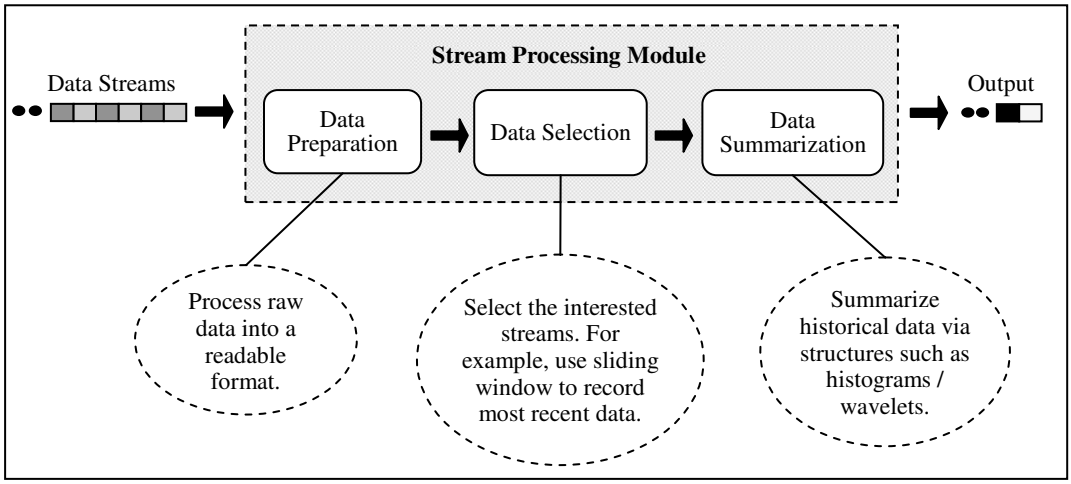


Figure 2.2 Stream Processing Module Architecture

2.2.1 Data Preparation

Data preparation includes data formatting and cleaning. Incoming data arrive in raw formats which in many cases cannot be efficiently handled by the DSMS. Hence, an efficient method must be used to transform the data into a format that can be rapidly processed by the remaining subsystems. Data cleaning

is necessary to reduce errors and computational overhead of the results. A typical data cleaning task would be to remove invalid data items (e.g., remove negative values when non-negative values are expected) using a simple filtering based method.

2.2.2 Data Selection

The primary goal of data selection is to reduce the total data size. Previous works have proposed the following three main methods to address this issue:

1. **Sliding window** – gives a window of size *L* (can be dynamic) to store *L* most recent data elements. The motivation behind this data structure is the assumption that recent data carry greater significance over old data. There are several applications which can fulfill this assumption such as monitoring and surveillance systems. Extensions of this sliding window structure have been proposed to store data at varying time granularity and length [7].
2. **Sampling** – assigns a probabilistic value to each data element under selection. The advantages of the sampling technique are that it can dramatically reduce the data size and provide a probabilistic error bound (Hoeffding bound [49]) for a given sampling rate. The disadvantage is that potentially important data items (e.g., outliers) may be missed and hence sampling may not be directly applicable to outlier detection tasks.
3. **Load shedding** – utilizes a similar scheme to sampling except that certain sequence sets of data elements are discarded. This approach has demonstrated its applicability in stream environment where the incoming data rate is bursty. Load shedding will selectively drop high rate streams and divert resources to higher priority tasks [29]. Load shedding shares the same drawback as sampling as it may remove some important anomalous data elements.

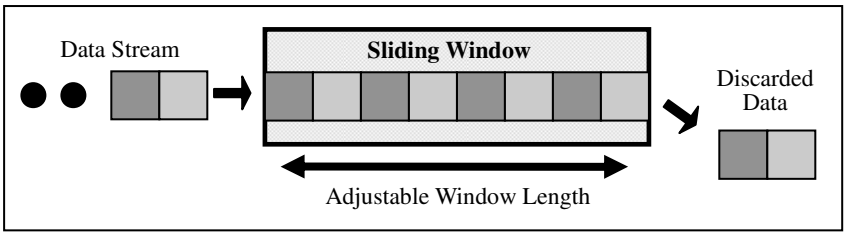


Figure 2.3 Adjustable Sliding Window

Due to the simplicity, generic nature, and wide applicability of the sliding window, this proposed research will focus on the development of algorithms under this data selection model. The sliding window (
Figure 2.3) is an adjustable structure that stores data elements of predefined length and order. The data element replacement policy and window adjustment will depend on the specific application. For example, a static length and First-In-First-Out replacement policy window may work well for non-prioritized streams, but may not be suitable for prioritized streams. Data selection must be able to mitigate a wide

spectrum of stream arrival characteristics with the limited space and computational resources.

2.2.3 Data Summarizations

The data summarization task is responsible for extracting (or summarizing) useful information about the selected data as it relates to the data stream task. Specifically, this step provides the necessary operation to efficiently update and maintain statistics about the overall data population. These abstractions can be provided through synopsis structures such as histograms, wavelets coefficient trees [60], etc. Some methods exist that can fulfill this requirement, especially ones developed from compression analysis, but these existing techniques must be adapted to meet the DSMS constraints. Therefore, it must minimally expend the system's computational and space resources. This goal can be attained done by employing incremental update techniques and allowing for higher representation error. It is clear at this point that the choice of a model is heavily dependent on the given task. For example, if outlier detection is primary aim then Wavelet Transform [21] may provide a better model than the Discrete Cosine Transform [73] since wavelets can more effectively represent singularities. There is a trend in the stream research community to develop highly specialized but efficient summarization techniques. For instance, DataSurg [71] has developed a specific modeling solution via Peano-trees [24] that can solve a limited set of query processing problems.

2.3 Mining Module

The mining module (Figure 2.4) will process the window records and synopsized information to determine the results of the assigned mining query. Because of the DSMS's highly restrictive constraints, different mining tasks will often necessitate specific mining approaches. Generalized mining approaches can seldom fulfill the constraints of the data stream environments. The mining tasks defined in the DSMS mining module are similar to the classical mining tasks but with the following key differences:

1. Employ iterative updates to the model.
2. Obtain approximate answers rather than determining exact solutions.
3. Integrate data order information.

The first two adaptations are done to achieve a single pass examination of the data, lower computational overhead, and reduce memory usage. At the conceptual level, iterative updates seem to garner a natural support from well-known classical mining algorithms such as ones based on expectation-maximization [61]. However, these methods assume that the working set of an upcoming algorithmic state is a superset of the current working set. Hence, data is only assumed to be added at any given future state but not removed as may be needed for stale/invalid items in a data stream. Approximate alternatives are also used to reduce the time complexity of computationally intensive algorithms. This approach can also be extended to generate adaptive accuracy guarantees by capitalizing upon information of currently available resources [30]. The last point of difference stems from the potential disassociation of actual data arrival

order and the data order for which is meaningful to the streaming application. For example, a traffic incident detection system requires observed speed measurements from several roadway sensors that are temporally sorted. However, due to the existence of differing sensor output policies and data loss, sensor data may not arrive in the expected ordering.

A data mining algorithm is composed of the following components: task, model structure, score function, search method, and data management [38]. For data streams, the model structure, search method, score function, and data management techniques may all need to be altered in order to fulfill the DSMS constraints. For some tasks, an adaptation to integrate efficient iterative updates and to generate approximate results may suffice the DSMS requirements. This process generally implies changes to the search method and data management components of the original offline mining algorithm. Excellent examples of these adaptations can be found in [25, 50, 62, 74]. However, for other mining tasks, dramatic changes to classical data mining algorithms must be performed to meet the stringent DSMS criteria. For example, adjusting model structure in order to cope with complex streams can lead to a new score function. As a consequence, the score function can enforce significant changes to the search and data management methods.

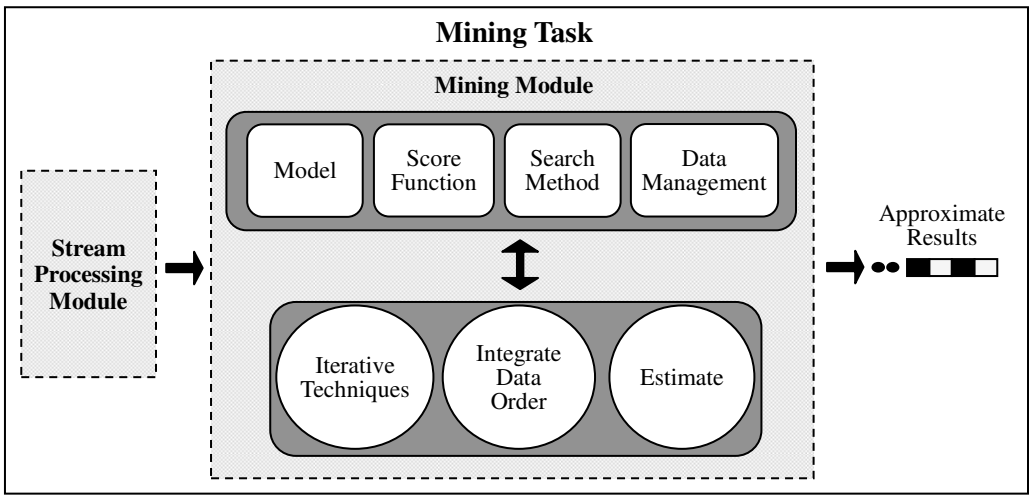


Figure 2.4 Mining Module Architecture

The mining component may need to address the issue of data uncertainty. Uncertainty may arise from the stream processing module, data, or mining algorithm itself. Errors accumulate at the stream processing module due to limitations such as finite window size and lossy data representation. Errors in data can arise due to noise, inadequate measurement precision, and low sensor reliability. The mining component may need to resolve errors that have been propagated in order to minimize the resultant errors.

Aside from the challenges mentioned above which are primarily derived from the DSMS constraints, a separate but crucial issue that must be addressed is the management of c*oncept drifts.* For several DSMS applications, incoming data streams will exhibit high amounts of changes in the underlying data generator.

The mining module must therefore recognize these changes and make the appropriate alteration to its model structure. For example, classification tasks can be greatly enhanced if it can detect the concept drifts and retrain with the new samples.

2.4 DSMS Applications

Early works in data stream primarily focused on the design and implementation of efficient storage and retrieval. Systems such as STREAM [9] and COUGAR [93] concentrated on the development of algorithms for the following query problems:

- Predefined query – issued prior to data arrival
- Ad-hoc query – issued online
- One-time query – invoked on a data stream snapshot
- Continuous query – continually evaluated against arriving data streams

The results of these early works have added significant advancements to the understanding of stream modeling, probabilistic bounded data selection, and efficient stream-based query processing [12, 13, 15, 16, 85]. The establishment of concrete models and availability of efficient algorithms later seeded interests in developing solutions to mining data streams. The implementations of these solutions can be scoped within the DSMS framework and categorized into the following classes: single task and multiple tasks based systems. Single task based systems are defined to generate the results of a single mining operation such as clustering or provide components (e.g., mining module) of the DSMS. It is no surprise then that the bulk of the stream mining works is composed of single task based systems. Multiple task based systems perform a multitude of tasks achieved through the compositions of single task based systems. The system may output a single or multiple result(s) generated from invocations of different mining operations. Because multiple single task based components must be managed, integration is an important issue that can involve significant technical investments in order to minimize the impact of increased requirement.

2.4.1 Single Task Based Systems

In single task based systems, a great deal of the attention was given to the issues of clustering and classification. One of the earliest approaches to clustering was proposed by Guha et al's [37] which employs a variant of the K-Medoid algorithm to data streams. The algorithm utilizes an incremental strategy to insert new data items and reweighs existing data to generate the updated centers. Guha et al's stream-based solution reduces the allowable search space for cluster construction. Due to limited memory, the proposed incremental approach necessarily restricts the assignment of sample points to their temporal ordering (i.e., arrival order). Unfortunately, this restriction can introduce optimums with higher residuals which thereby increase the rate of false dismissals.

Recent improvements to Guha et al's algorithm extended its applicability to the sliding window model and provided potential improvements in reducing the time complexity [14, 22]. However, because these extensions are based on the original K-Medoid algorithm, they inherit a fundamental problem – a constant K may not be able to appropriately model a dynamically changing data stream. Aggarwal et al. [7] developed the CluStream algorithm in an attempt to mitigate this issue by automatically defining K. Initially, K is assigned to the largest possible value that will fit into main memory to provide a highly granular description of the data. These fine-grained clusters can then be merged to give various summarizations of the stream at different cluster resolutions. CluStream separated the clustering task into a pair of online and offline components. The online step maintains a set of small dense clusters (called micro-clusters) which represent the most recent data. The micro-clusters are dynamically created, merged, and removed (based on oldest timestamp) as new data arrive. Hence, the total number of clusters can dynamically adapt to the stream's changing behavior. The deleted micro-clusters are recorded into a secondary storage which can later be retrieved for further analysis. The offline task is responsible for storing and managing necessary statistics of the removed clusters. These stored clusters can later be merged and segregated to give various synoptic views. Another clustering paradigm employs the use of convex hulls in order to reduce the amount data processing [47]. By modeling the shape of the clusters via convex hulls, data points that reside within the cluster need not to be included in the maintenance procedure. However, these data points may be used to update statistics such as the cluster weight. One application of this approach to detect interesting patterns for moving objects.

One of the most commonly referred stream classification algorithm is the Very Fast Decision Tree (VFDT) [25]. It induces an incrementally growing tree and incorporates the Hoeffding bound to provide level splitting criterion with probabilistic error bound [49]. Aggarwal et al. demonstrated a modified version of the CluStream algorithm to support nearest neighbor on-demand classification [8]. Under this approach, the micro-clusters with an appropriate distance criterion are assumed to be assigned with a set of labels. The data samples are then labeled with the class of the nearest neighbor micro-cluster. A useful application of classification is to automate certain surveillance tasks. For example, real-time system has been developed to classify aerial images for geographic feature (e.g., mountains, lakes, etc) labeling [67, 71, 90]. The system transforms the areal images (via Peano space-filling curve [75]) into a quad-tree [78] like structure and performs classification in various regions of the transformed image.

2.4.2 Multiple Tasks Based Systems

Existing multiple tasks based systems are frequently designed to operate within a particular application scenario. An example of such a system is MobiMine [53] which provide real-time stock alerts to mobile devices. MobiMine employs a client-server architecture that automatically provides information of abnormally behaving stocks that are of potential interest to the users. Initially, the user produces a list of interested stock categories to the central server. The server continuously monitors all available stocks from several sources (e.g., web pages, RSS feeds) and performs online outlier detection and clustering to obtain stocks that are behaving abnormally in groups. The employed outlier detection and clustering

schemes are adaptations of existing single task based systems. If a significant number of members of an abnormal group belong to one of the user's list of interested categories, then the stocks are forwarded to user's mobile device. MobiMine also provides a method for compressing the amount of transferred data by applying Fourier transforms and retaining a subset of the coefficients.

VEDAS [52] is multiple tasks based system that monitors vehicles in real-time. The system detects high risk driving behavior (e.g., drunk driving) and abnormal mechanical states. Installed in each vehicle is a general processing unit which collects data from the vehicle's built-in sensors (e.g., engine thermometer, air mass sensor). The vehicle features are continuously mined to alert any abnormality to a central server. The mining operations include incremental principal component analysis to reduce the data dimension. This dimensionally reduced dataset is then used to seek for abnormal patterns by employing an online clustering algorithm and statistical tests. A polygon based clustering (similar to Convex hull approach discussed previously) is developed to rapidly model the vehicle's driving characteristics. If the vehicle's current driving state does match its expected model, then a statistical significance test is performed on the outlying dataset. A statistically significant state is reported to the central server for evaluation.

The EVE [87] system provides an infrastructure by which different and multiple mining tasks can be implemented for use in an on-board satellite environment. Data from the satellite sensors arrive as streams, online mining tasks are performed on-board, and the results forwarded to the ground station. EVE employs an object oriented based paradigm to implement the DSMS and introduces a set of types to represent certain operations and mining tasks. A major contribution of the EVE system is the provision of an event management system that can successfully integrate the various abstracted mining objects within the highly resource constrained environment of a satellite.

Although the above stream systems primarily deal with real-time mining, the following system employs a dual focus of real-time mining and offline analysis. Genie of the Net [68] is a multiple tasks based system that supports the discovery of interesting context states for cyclists. To target for particularly interesting states, the data of an active cyclist (e.g., heart rate, speed, elevation, etc) are recorded and analyzed offline with traditional mining techniques. The real-time mining components of the system serve to produce sufficient data features for the offline analysis. For example, the cyclist's current elevation is determined using a variety of online classification techniques. The elevation attribute is then used by the offline analysis module to provide useful context states that can facilitate the cyclist's exercising regime.

2.5 Integration of the Density Estimator

This section describes the general approach and operations of a stream density estimator. The results of the density estimates can be employed by various mining algorithms (implemented in the mining module) as a venue for deriving distributional information. However, density estimation can also serve as the mining component to tasks such as exploratory analysis. As a consequence, the density estimator can subsume the role of data summarization within the stream processing module or the mining module in the

DSMS.

2.5.1 Supported Operations

The density estimator should support the following set of generic operations:

- Data insertion
- Data deletion
- Window shaping
- Density query estimation

Data insertion corresponds to the model update of a new and recent item that is added to the sliding window. Conversely, data deletion handles the changes to the model caused by a removal of an old item within the sliding window. These operations may be sequentially and continuously invoked due to the potentially high arrival rates of the data streams. Hence, providing efficient methods for handling data insertions and deletions is a key goal in the development of the density estimator. Window shaping refers to the of ad-hoc reassignment of the sliding window size. In situations where the temporal range of interest shifts, the window size will need to be quickly adjusted to correctly reflect the new interest. For example, in monitoring highway traffic, a default window size of the last 1-hour of speed observations may be used under normal conditions. However, when an incident occurs, the operators may prefer to view only the last 10-minutes of data to reduce the polluting affects of old and non-incident related observations. Hence, window shaping can be described as the model updates caused by the elimination of a certain set of old items.

2.5.2 Density Estimation Approach

The approach for providing a density estimator follows a similar process to the mining module in Section 2.3. The density estimator employs a model, score function, search algorithm, and management strategy to generate the density value of a particular query point. Furthermore, the component supports insertion, deletion, and window shaping operations via iterative approaches. Density estimates are generated as approximate results to be utilized as inputs to the mining module or serve as the mining module in the DSMS. Figure 2.5 depicts the density estimation approach and integration into the DSMS.

Form Selection

The model for density estimation is expressed by three basic forms: parametric, semiparametric, and nonparametric. The differences in these forms are the assumptions imposed on the underlying distribution. The parametric technique assumes that the structure of the distribution is known which provides for the fastest convergence rate. However, there is a greater risk of error if the assumed form does not match the

actual distribution. The semiparametric approach integrates a similar a priori form assumption with some additional flexibility. The flexibility is generally achieved by summing a set of parametric forms such as the case of the Gaussian mixture. The nonparametric technique solves the problem of erroneously choosing an a priori structure by allowing the "data to speak for themselves". This modeling approach is well-suited for data streams since the density functions are often unknown and dynamic. The nonparametric technique offers a slower convergence rate than the parametric approach; however, this drawback can be remedied by exploiting the data stream's large sample size.

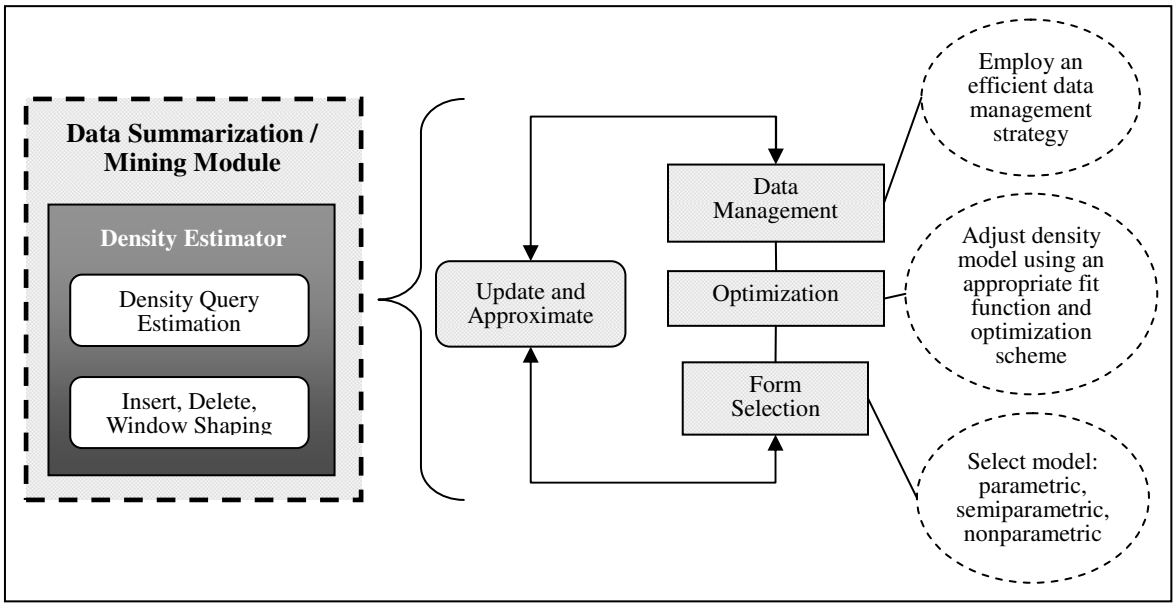


Figure 2.5 Density Estimation Approach and Integration into the DSMS

Optimization

The task of the optimization module is to fit the selected form to the underlying density function. In order to achieve this task, an appropriate measure is chosen which quantifies the model's fit to the true density structure. In the parametric and semiparametric cases, this measure can be the maximum likelihood which maximizes the probability of the observed data. The nonparametric approach may employ the asymptotic mean integrated squared error (AMISE) to generate an appropriate smoothing parameter to optimize the expected accuracy. A variety of optimization strategies are available which include iterative techniques and closed-form solutions (e.g., sufficient statistics for parametric models). In the nonparametric case, an accurate and dynamic smoothing parameter can be obtained via an exhaustive offline multi-phase estimation process [82]. However, such a method is not readily applicable to the resource constraint environment of a data stream. In fact, many of the optimization strategies assume that processing are done offline with sufficiently large resources. To better suit the needs of the data stream environment, this work proposes to develop fast, approximate, and incremental algorithms in order to construct a viable nonparametric stream-based density estimator.

Data Management

The data management component is responsible for providing the appropriate data structures and operations to facilitate compliance to the constraints of the data stream application. To enhance retrieval performance, indexes can be applied with some modifications to limit the update cost and space usage. Another point of consideration is that in data streams it can be useful to consider the actual complexity of an operation rather than its order. This is due to the fact that in real-time scenarios, a constant factor reduction in computation is highly pertinent when time is a precious commodity.

Density Estimation Model: Nonparametric Kernel Density Estimator

In real-world scenarios, the PDF is usually unknown and therefore its form (structure) must be estimated. An effective technique to estimate an unknown probability density function is the nonparametric kernel density estimate (KDE). KDE possesses several advantages 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 [80, 82]. Hence, this work focuses on the development of effective and efficient KDE approaches to model the density of data streams.

The standard formulation of the univariate KDE is given as follows: for n independent and identically distributed sample points (i.e., kernel centers) $x_1..x_n$ with corresponding weights $w_1..w_n$, bandwidth h, and a kernel function $K(\cdot)$, the kernel density estimate is

Equation 2.1

$$f_{KDE}(x) = \frac{1}{\sum_1^n w_i} \sum_{i=1}^{n} \frac{w_i}{h} K\left(\frac{x-x_i}{h}\right)$$

where $\int K(t)dt = 1$.

The density estimate at 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 since 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 portion of symmetric "bumps" that are centered on each data sample.

2.5.3 Existing Density Estimators

Early efforts in computationally tractable KDEs can be found in the domain of *offline* analysis. Zhang *et al.* proposed a method to maintain a space efficient KDE by using the CF-tree [96] to cluster a set of kernels into a single kernel known as the CF-Kernel [97]. The CF-Kernel employs a global bandwidth

which can lead to oversmoothing and loss of local density information. Gray *et al.* proposed a kernel space partitioning technique by utilizing a KD-Tree and bounded support kernels to offline datasets [34]. The KD-Tree reduces computations by effectively pruning kernels, which do not contribute to the density query.

Several recent works have been proposed for the online management of KDE. These online techniques can be classified into the following three categories:

1. **Grid-based KDE** – provides a uniform and discretized representation of the kernel points
2. **Sample-based KDE** – employs a sampling methodology on the data stream to reduce the total kernel size
3. **Cluster-based KDE** – utilizes kernel merging techniques to maintain a fixed storage and minimize the kernel merge error

Grid-based and sample-based

Aggarwal proposed a framework to capture the structural evolution of data streams by using a grid-based KDE [5]. The kernels are summarized in a multidimensional grid where a common bandwidth is employed for each dimension. Concepts of forward and reverse density profiles are introduced to discover the occurrences of concept drifts. Subramaniam *et al*. proposed an approach for outlier detection in sensor networks by modeling the densities of node observations [86]. Their scheme employs a uniformly sampled sequence-based sliding window to summarize the current set of kernels. A global bandwidth is applied to the sampled set of kernels. Wegman *et al.* introduced an online KDE for the analysis of Internet traffic [91]. They suggested the use of a sequence-based and exponentially aging sliding window to accommodate a fixed storage environment. To derive estimates, a single bandwidth KDE is employed on the sliding window.

Cluster-based

Zhou *et al.* introduced the M-Kernel, a cluster-based KDE maintenance strategy which performs numerically-based kernel mergers under a fading window [98]. The merging algorithm combines two kernels which produces the minimum integrated absolute error between the original pair and the merged result. This scheme allows for a fixed memory representation of the kernels. However, under the M-Kernel, the consistency of the estimate is not guaranteed and the approach can exhibit high update costs due to its numerical-based merging strategy. In a similar vein, Heinz *et al.* proposed a constant time pair-wise merging technique that guarantees consistency [43]. Their kernel method employs a single bandwidth that has been shown to be effective in approximating the classical KDE.

Multiple query processing

Efficient estimation of multiple density points have been proposed for offline KDE techniques. These methods operate on a transformed estimation problem expressed as a convolution of the kernel weight

and kernel influence function [89]. The convolution is efficiently computed via orthogonal series techniques such as discrete Fourier transform or fast Fourier transform [26, 63]. However, estimating multiple points through convolution-based approaches are limited in two critical ways: (1) a grid-based KDE must be employed; and (2) orthogonal series computation can only be applied when the underlying KDE utilizes a global bandwidth [89]. These two drawbacks reduce the estimation quality of complex distributions due to the uniform space partitioning of the grid-based method and the global bandwidth's inability to effectively model local structures. For the stream-based estimators, multiple queries are resolved by invoking several single query estimates in first-in-first-out order. These multiple invocations can result in significant computational overhead due to the redundant searches of (potential non-contributing) kernel objects and suboptimal query order. For data streams, any excess time allocated to computations can result in denials of much needed results to the application. Due to the above issues, this work develops efficient algorithms for the LR-KDE that can rapidly estimate multiple density queries.

Technique		**Single Query Cost (Worst Case)**	**Multi Query Cost (Worst Case)**	**Update Cost**	**Bandwidth Strategy**	**Asymptotic Consistency**	**Time-based Window**
Grid-based	[5]	$O(M)$	$O(\|Q\|M)$	$O(M)$	Global	Yes	Yes
Sample-based	[86], [91]	$O(M)$	$O(\|Q\|M)$	$O(M)$	Global	Yes	Yes
Cluster-based	Heinz [43]	$O(M)$	$O(\|Q\|M)$	$O(M)$	Global	Yes	No
Cluster-based	M-Kernel [98]	$O(M)$	$O(\|Q\|M)$	$O(M)$	Variable	No	No

Table 2.1 Summary of stream-based KDE techniques, M is the number of maintained kernels and $|Q|$ is the density query size

Table 2.1 provides a summary of the key characteristics of current stream-based KDEs. Most KDE approaches employ single a bandwidth strategy hence they cannot accurately estimate the stream's local features. The M-Kernel, although it applies a variable bandwidth approach, it is not assured to be asymptotically consistent. As a consequence, the M-Kernel is not guaranteed to converge as the sample size increases.

Online histograms have also been proposed in the field of database optimization to provide query selectivity estimates and approximate queries [51]. Some online histograms include dynamic quantiles [33], equidepth histograms [32], and V-optimal histograms [35]. Due to the histogram's inherent discontinuities and slower convergence, the histogram may not be suited for the tasks of stream analysis [82].

2.6 Conclusion

This chapter gives a background on data stream mining within the context of the data stream mining system framework. The framework combines the foundational elements of existing stream-based mining solutions. Additionally, the chapter provides the requirements, research issues, and a guideline for designing a viable stream density estimator.

Chapter 3. Local Region Based Kernel Density Estimator

This chapter describes the proposed nonparametric Local Region Kernel Density Estimator (LR-KDE). Probability density function estimation is a fundamental component in several stream mining tasks such as outlier detection and classification. As such, accurate estimates of the PDF are essential to reducing the uncertainties and errors associated with mining results. The nonparametric adaptive kernel density estimator (AKDE) provides accurate, robust, and asymptotically consistent estimates of a PDF. However, due to AKDE's extensive computational requirements, it cannot be directly applied to the data stream environment. This chapter describes the development of an AKDE approximation approach that heeds the constraints of the data stream environment and supports efficient processing of multiple point density queries. To this end, this work proposes (1) the concept of local regions to provide a partition-based variable bandwidth to capture local density structures and enhance estimation quality; (2) a suite of linear pass methods to construct the local regions and kernel objects online; (3) a multiple density point evaluation algorithm that reduces the search costs; (4) a set of approximate interpolation based techniques to increase the throughput of multiple density point query processing; and (5) a fixed-size memory time-based sliding window which updates the kernel objects in linear time. Analyses of the estimator's asymptotic consistency and computational costs are also provided. Comprehensive experiments were conducted with real-world and synthetic datasets to validate the effectiveness and efficiency of the approach.

The proceeding sections are organized as follows. Section 3.1 describes the motivation and contribution of this chapter. Section 3.2 provides the theoretical preliminary. Section 3.3 details our proposed LR-KDE approach. Section 3.5 presents the analyses on LR-KDE's computational complexity and asymptotic consistency. Section 3.6 demonstrates empirical results and validation. Section 3.7 gives the conclusion.

3.1 Motivation

As mentioned in Chapter 2, the nonparametric kernel density estimator (KDE) provides the underpinning for several stream mining algorithms. Examples of mining algorithms that employ the KDE include outlier detection of a sensor network [86], concept drift analysis of spatiotemporal streams [5], and pattern discovery within Internet traffic [91]. One could further the efficacy of the KDE by enabling queries for an explicit time range [6]. The extension would be able to respond to questions such as, "What is the distribution of telnet connections within the last hour?" "How have the roadway's speed and volume distributional patterns changed in the past two hours since the snow began?" The extension of an explicit time range is naturally modeled by a time-based window. Hence, it can be seen that providing density estimates over a time-based sliding window will serve as a valuable tool in the study of data streams. Another desired estimation feature is the provision of a rapid approach for generating densities at multiple points. As in the examples above, the outlier detection strategy in [86] approximates the distance-based outlier measure by invoking multiple KDE queries within an interval; in the visualization work of [91], multiple queries are utilized to produce estimates at regular grid points; and concept drift detection

estimates densities at multiple points to compute a particular divergence score (e.g., Kullback-Leibler distance [55]) between the current and past densities. A slight variant to the above scenario is the case of data stream monitoring where several agent processes may query various parts of data distributions simultaneously. Because existing stream-based KDEs are designed to generate a single query estimate, processing multiple single queries within these estimators can lead to certain operational redundancies that degrade query throughput. Hence, new estimation techniques must be developed which can effectively remove these redundancies to efficiently process multiple and concurrent density queries. Emphasis on the most recent data implies the need for the stream mining techniques to furnish results in real-time. Additionally, practical constraints dictate that the techniques possess finite memory and perform at most a linear pass on the data elements [11, 64]. Meeting and balancing these requirements is a key focus of this work.

3.1.1 Need for an Adaptive KDE Approach

The accuracy of the KDE does not significantly depend on the choice of kernel function $K(\cdot)$, but rather on the selection of h [80]. The bandwidth h is regarded as a smoothing parameter: a high h can generate a smooth shape density whereas a low h tends to provide an undersmoothed but potentially more accurate estimate. The drawback of the KDE formulation is the requirement of assigning a global bandwidth. Due to the existence of local features, a single global bandwidth may not be sufficient to model complex density structures (e.g., multimodal distributions). When the global bandwidth KDE technique is used as the core step in mining tasks (e.g., outlier detection), the generated results can be misleading and potentially disastrous for mission critical applications (e.g., military sensor surveillance). To overcome this problem, this paper proposes the use of an adaptive KDE (AKDE) to improve the estimation accuracy of local features within data streams.

The AKDE is a variable bandwidth technique, which has been shown to be effective in capturing local density features [77, 80, 82]. The general formulation of the AKDE is as follows:

Equation 3.1

$$f_{AKDE}(x) = \frac{1}{\sum_1^n w_i} \sum_{i=1}^{n} \frac{w_i}{H(x_i)} K\left(\frac{x - x_i}{H(x_i)}\right), \; H(x_i) \propto f(x_i)^{-1}$$

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

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 AKDE characteristic is consistent with the real-world observation: inherent local features tend to occur in regions of high probabilities whereas noise-induced local features occur in areas of low probabilities. This adaptive scheme enables the AKDE to provide superior estimation accuracy over the classical KDE.

Although the AKDE can produce superior estimation quality to the classical KDE, its computational cost

($O(n^2)$) far exceeds the traditional 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 classical 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 because $H(\cdot)$ is computed at least once for each sample point. This evaluation approach cannot be applied to data streams since it clearly infringes upon the linear pass restriction. As a further challenge, the AKDE is formalized as a technique which only generates a single density estimate. Hence, multiple queries must be submitted in sequence which results in several redundant computations and further throughput degradation. Because the query points are determined ahead of time either by the mining algorithm or application usage (e.g., multiple users submitting queries), new query processing methods can be constructed to exploit this input availability in order to reduce redundant computations and improve overall query throughput. Due to the extensive computational costs involved with AKDE, there are currently no works that provide adaptive kernel density estimates for the data stream environment.

A crucial property provided by the KDE is asymptotic consistency, that is, as the sample size approaches infinity, the KDE converges to the true density with converging probability of 1. This KDE property is conducive to the data stream environment because the unbounded sample size provided by the data stream can improve the KDE's accuracy. Therefore, to enable AKDE to be a viable tool in stream analysis, a new online technique must be developed that is both efficient and asymptotically consistent.

3.1.2 Contribution

This chapter tackles the issue of developing efficient and asymptotically consistent AKDE capable of rapidly generating multiple density estimates over data streams. To the best of our knowledge, this work is the first attempt that addresses the issue of constructing a linear time AKDE for the data stream environment that supports efficient multiple query evaluations. To that end, we propose the online Local Region KDE (LR-KDE), an adaptive kernel density estimation framework for processing univariate data streams. The major components of the proposed framework are (1) a new partition-based variable bandwidth technique to capture local density structures and to enhance estimation quality, (2) a suite of *linear pass* methods to construct local regions and its corresponding kernel objects online, (3) a multiple density point evaluation algorithm that reduces the costs of local region and kernel object searches, (4) a set of interpolation based techniques which approximate and further increases the throughput of processing multiple density point queries, and (5) a fixed-size memory time-based sliding window which updates and expires the kernel objects in linear time. We also analyze the asymptotic costs and consistency property of the online LR-KDE. Extensive experiments were conducted to demonstrate the effectiveness and efficiency of the approach.

The proposed LR-KDE differentiates itself from existing estimators (see Section 2.5.3 for survey) in the following aspects:

1. **Local feature estimation** – models local density features via partition-based bandwidth to improve estimation quality
2. **Consistency** – assures asymptotic consistency under the proposed variable bandwidth strategy
3. **Multi-query optimization** – generates multiple density estimates that eliminates redundant query operations and enhances overall query throughput with a worst-case cost of $O(|Q|M)$ where $|Q|$ is the size of the query set and M the number of maintained kernel objects
4. **Linear pass processing** – employs $O(M)$ algorithms to process kernel updates and density queries
5. **Time-based window** – provides a fixed-size time-based sliding window

3.2 Local Region Concept

The problem addressed in this chapter is as follows: *Given a data stream* $D = \{x^{(1)}, x^{(i)}, .., x^{(n)}\}$ *where* $x^{(i)}$ *is a real-valued scalar (i.e., sample point) and each* $x^{(i)}$ *is associated with a timestamp* $\tau(x^{(i)}) \in [\tau_{min}, \tau_{max}]$ *generate and maintain an adaptive kernel density estimator* $\hat{f}_{AKDE}(\cdot)$ *of* S*. The storage, update, and query costs of* $\hat{f}_{AKDE}(\cdot)$ *should be at most* $O(M)$*, where* M *is a constant and* $M \ll \|S\| = n$*.*

With respect to stream applications, one of the fundamental issues of the AKDE is its high computational cost for determining the bandwidth, $H(\cdot)$ (Equation 3.1). To reduce this computational requirement, a bandwidth approximation technique is proposed for the AKDE. Let $T = \{x_i : x_i \leq x_{i+1}, 1 \leq i \leq n, x_i \in S\}$ be an ordered representation of the kernels in D. Define the *relative density variance*, $R(k, l)$, as the sample variance of the set of estimated densities at $x_k..x_l \in T$, where $1 \leq k \leq l \leq n$. The bandwidth approximation procedure is given as follows:

1. Partition T into Q continuous and disjoint *local regions* (i.e., intervals) such that each local region's $R(\cdot,\cdot)$ value is minimized
2. For each local region, assign a bandwidth that is unique to its constituent kernels

The above scheme serves to capture the local densities within the total span of the distribution. The obtained approximation is consistent with the structure of AKDE's bandwidth i.e., similar bandwidth values are assigned to kernels of similar densities. Hence, the local regions can be seen as a piecewise constant representation of the structure of $H(\cdot)$.

Two design challenges exist in applying the above approximation approach: (1) the efficient derivation of the relative density variance; and (2) the development of a technique for local region identification. In the following, the Pair-wise Adjacent Distance Uniformity theorem is introduced to provide an efficient venue for estimating the density variance. The theorem is followed by the definition of an optimization problem for the task of identifying the local regions.

3.2.1 Efficient Estimation of the Relative Density Variance

As previously defined, local regions provide a total and disjoint partitioning of the kernel domain which minimizes the intra-variance of the density estimates. A unique bandwidth is assigned to each local region based on the regional kernel characteristics. If each kernel is given its own unique local region, then the local region based KDE (with the appropriate bandwidth function) is a reformulation of the traditional AKDE. However, if the number of the local regions is less than the number of kernels, then the local region KDE is an approximation of the AKDE. The problem now is to derive a method which can efficiently deduce the density estimate variance for a given range of kernels. One possible approach is to estimate the densities via the traditional KDE approach. However, this poses the same computational issue as the AKDE. An alternative and more viable solution is to employ the pair-wise and adjacent kernel distances to derive relevant properties of the density variance. To that endeavor, the Pair-wise Adjacent Distance Uniformity theorem is proposed.

Let G be a set of ordered and identically weighted kernels whose pair-wise and adjacent distance variance is zero, then under certain conditions, it can be shown that the densities at the kernel centers (under a global bandwidth KDE) in G are uniform. The significance of this theorem is that it provides a venue of estimating the density variance through information of the kernels' pair-wise and adjacent distances (PAD). As a result, computationally tractable bandwidth approximations can be developed from the kernels' PAD information. The proof of the Pair-wise Adjacent Distance Uniformity theorem is as follows:

Theorem 3.1 Pair-wise Adjacent Distance Uniformity: Let $V = \{x_i : x_i \leq x_{i+1}\ and\ 1 \leq i \leq n\}$ be a set of sorted kernel centers associated to a bounded and radially-symmetric kernel function with uniform weight and bandwidth. Furthermore, let the sorted kernels be adjacently equidistant such that $|x_{i+1} - x_i| = |x_{j+1} - x_j|\ \forall i, j$. Suppose $G = \{x_k : x_1 + bandwidth \leq x_k \leq x_n - bandwidth\}$. Then for the kernel density estimate, $\hat{f}(x)$, the following property must hold: $\hat{f}(x_k) = \hat{f}(x_l)\ \forall\ x_k, x_l \in G$.

Proof. Let $x_k, x_l \in V$ be any two kernel centers and define a set $I_x = \{x_i : |x_i - x| < bandwidth\}$. I_x is the set of kernel centers for which their corresponding bandwidths intersect x, thus I_x possess all the kernels that contribute non-zero values to the kernel density estimate, $\hat{f}(x)$. Consider I_{x_k} and I_{x_l} and without loss of generality choose a kernel center, $\alpha \leq x_k$, from I_{x_k}. Because all the kernel centers within G are adjacently equidistant, there must exist an element, $\beta \leq x_l$, from I_{x_l} such that $x_k - \alpha = x_l - \beta$. Since the kernels are radially-symmetric, equal bandwidth, and identically weighted, the contribution of α to $\hat{f}(x_k)$ is equal to the contribution of β to $\hat{f}(x_l)$. For any chosen α in I_{x_k}, there exists a corresponding β in I_{x_l} for which their contributions are equal. This relationship also holds for any β in I_{x_l} corresponding to an α in I_{x_k}. Hence, there is a one-to-one relationship between α in I_{x_k} and β in I_{x_l}, which implies that the sum of the kernel contributions of I_{x_k} to $\hat{f}(x_k)$ and I_{x_l} to $\hat{f}(x_l)$ are equal. Therefore, $\hat{f}(x_k) = \hat{f}(x_l)\ \forall\ x_k, x_l \in G$.

3.2.2 Optimization Problem for Local Region Identification

Leveraging upon Theorem 3.1, identification of the local regions can be achieved by minimizing the variance of the kernels' pair-wise and adjacent distance values. In the following, an optimization problem is established to identify local regions based on the kernels' PAD information. Assume that a local region, L, possesses a set of kernels (centers), $y_j..y_m$, where $y_j \leq y_{j+1} \leq y_m$ for $1 \leq j \leq m \leq n$, and n is the total number of kernels. Set L's i^{th} adjacent distance to be $L_d(i) = y_{j+i+1} - y_{j+i}$, and define the variance of L's kernel pair-wise and adjacent distances as follows:

Equation 3.2

$$var(L_d) = \frac{1}{\|L\|-1}\sum_{i=1}^{\|L\|-1}\left(L_d(i) - \frac{\sum_1^{\|L\|-1} L_d(i)}{\|L\|-1}\right)^2$$

where $\|L\|$ is the number of kernels in L.

In practice, the density estimate makes use of arbitrarily weighted kernels. Hence, consideration of varying weights must be made in the local region formulation. Let $z_1..z_n$ be a set of sorted kernel centers assigned to a bounded, radially-symmetric, and equal bandwidth kernel function, such that $|z_{i+1} - z_i| = |z_{j+1} - z_j| \; \forall i, j$. Let $w(z_i)$ be the non-negative weight of kernel center z_i, and $\hat{f}(z_i)$ be the density estimate at z_i. From the KDE definition, $\hat{f}(z_i) \propto w(z_i) \Longrightarrow var\left(\hat{f}(Z)\right) \propto var\big(w(Z)\big)$.

Therefore, minimizing $var\left(\hat{f}(Z)\right)$ is equivalent to minimizing $var\big(w(Z)\big)$. Let $L_{kw}(i)$ be the weight of kernel y_{j+i} in L, and define L's kernel weight variance, $var(L_{kw})$, as follows:

Equation 3.3

$$var(L_{kw}) = \frac{1}{\|L\|}\sum_{i=1}^{\|L\|}\left(L_{kw}(i) - \frac{\sum_1^{\|L\|} L_{kw}(i)}{\|L\|}\right)^2$$

Using the local region PAD (Equation 3.2) and the kernel weight variance (Equation 3.3) formulae, partition the entire sample points $y_1..y_n$ into Q disjoint local regions by minimizing the aggregate variance of all L_d and L_{kw}:

Equation 3.4

$$\text{Min } \sum_{i=1}^{Q}\|L\|\left(var\left(L_d^{(i)}\right) + \mu \cdot var\left(L_{kw}^{(i)}\right)\right)$$

where μ is the weight assigned to L's kernel weight variance.

In summary, the solution to the above local region optimization problem generates local regions with minimum overall intra-density variation. Moreover, the problem is solved by exclusively employing the kernels' PAD and weight information which are amenable to efficient implementations.

3.2.3 Connection to Clustering

The concept of local region is related to the notion of clustering. To model clusters, an optimization task is performed by assigning data samples to a set of classes which maximizes the intra-class similarity and inter-class discrepancy. The employed similarity/discrepancy metrics vary across applications and different optimization strategies are utilized based on the chosen discrepancy score and domain. Section 2.4.1 provides an overview of related stream-based clustering methods. These clustering algorithms can be viewed as data summarization techniques where the summarized results (i.e., identified classes) become proxies to an analytical task. Within the context of this work, local regions are classes of kernel objects which have minimum estimated density variance. A direct optimization strategy based on this discrepancy metric would incur $O(M^2)$ time. However, a novel discrepancy score based on Theorem 3.1 reduces this optimization task to $O(M)$. Within the domain of stream-based kernel density estimation, this is the first known application of clustering to *efficiently* produce a set of local bandwidths that *effectively* approximates the AKDE and guarantees estimation consistency.

3.3 Proposed Approach: Online Local Region KDE

The local region identification problem can be solved via dynamic programming techniques in time $O(n^2Q)$ where Q is the number of local regions; however, this solution exceeds the constraints of the data stream problem. Therefore, an incremental local region identification technique is proposed. The technique employs a locally optimal decision strategy to reduce the identification task to $O(nQ)$. As a result, the online Local Region KDE (LR-KDE) is proposed for the efficient generation of density estimates in data streams. An overview of the LR-KDE architecture is given in Section 4.1. Detailed descriptions of the local region and kernel constructions are provided in Sections 4.2 and 4.3. Lastly, the single query density estimation algorithm is described in Section 4.4.

3.3.1 Online LR-KDE Overview

The online LR-KDE is composed of the following two primary components:

1. **Local region management:** Identifies and manages local regions by employing a locally optimal dynamic binning method for the entire set of kernels.
2. **Kernel maintenance:** Updates the kernels with new data points in a fixed-size memory environment. Maintains and provides density evaluation of kernels whose time stamps are within $[\tau_{min}, \tau_{max}]$.

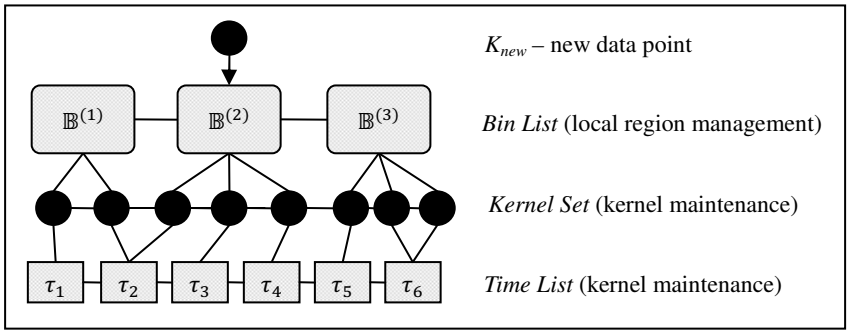


Figure 3.1 Online Local Region KDE

Figure 3.1 illustrates the LR-KDE approach. An online binning method is applied on the set of all kernels where each bin represents a local region. The bins are maintained in the data structure, *bin list*, which can store at most Q number of bins. The *kernel set* manages and organizes *all* kernels in a sorted queue ordered by their centers. A maximum of $M \geq Q$ kernels is maintained in the kernel set. The *time list* structure is a sorted queue of kernel arrival times. The objective of the *time list* is to model a time-based sliding window (using the FIFO policy) and to support efficient operations of kernel expirations and insertions. The following is a summary of the main operations for inserting a new data point K_{new}:

1. **Bin list update:** Find the bin whose interval (bounded by its corresponding minimum and maximum kernels) intersects K_{new}; add a reference to K_{new} to the bin's set of kernels; and forward K_{new} to the *kernel set* list. If no intersecting bin exists, create a new bin for K_{new} and (possibly) merge two adjacent bins to maintain a maximum of Q bins.
2. **Kernel set insertion:** Insert K_{new} into position by searching the *kernel set*. If after the insert, the kernel set size is greater than M, then merge two kernels of the kernel set which minimizes the overall accuracy loss.
3. **Time list synchronization:** After K_{new} is added to the *kernel set*, its arrival time is appended to the tail of the *time list*. If a kernel merger occurs, then its corresponding timestamps will also need to be updated in the *time list*. The head of the *time list* will also need to be verified against τ_{min} in order to remove expired kernels.

3.3.2 Local Region Management

The following proposes an incremental bin management approach for the identification of local regions. A formal definition of the bin and its property are given as follows:

Bin Vector –a bin vector, $\mathbb{B}$, is defined as follows:

Equation 3.5

$$\mathbb{B} = [b_{ks}, b_{ss}, b_{ws}, b_{wss}, b_{wcs}, b_{wcss}, b_{sp}]$$

b_{ks} is the set of kernels in $\mathbb{B}$ sorted by their kernel centers. The following feature definitions refer to set b_{ks}. b_{ss} denotes the squared sum of the kernel centers. b_{ws} gives the sum of the kernel weights. b_{wss} yields the squared sum of the kernel weights. b_{wcs} indicates the sum of the weighted kernel centers. b_{wcss} provides the weighted squared sum of the kernel centers. Let $x_1..x_{b_c}$ be the sorted kernel centers in b_{ks}, then the sum of adjacent kernel centers product is $b_{sp} = \sum_{i=1}^{b_c-1} x_i x_{i+1}$.

The bin vector possesses the additivity property [96]. This property allows the combining of bins in constant time. Let $\mathbb{B}^{(1)}$ and $\mathbb{B}^{(2)}$ be a pair of disjoint bins, $\max\left(b_{ks}^{(1)}\right)$ = the maximum kernel in $b_{ks}^{(1)}$, and $\min\left(b_{ks}^{(2)}\right)$ = minimum kernel in $b_{ks}^{(2)}$. If $\max\left(b_{ks}^{(1)}\right) < \min\left(b_{ks}^{(1)}\right)$, then the sum of bin vectors $\mathbb{B}^{(1)} + \mathbb{B}^{(2)}$ is:

Equation 3.6

$$\mathbb{B}^{(1)} + \mathbb{B}^{(2)} = \begin{bmatrix} b_{ks}^{(1)} \cup b_{ks}^{(2)},\ b_{ss}^{(1)} + b_{ss}^{(2)},\ b_{ws}^{(1)} + b_{ws}^{(2)},\ b_{wss}^{(1)} + b_{wss}^{(2)},\ b_{wcs}^{(1)} \\ +b_{wcs}^{(2)},\ b_{wcss}^{(1)} + b_{wcss}^{(2)},\ b_{sp}^{(1)} + b_{sp}^{(2)} + \max\left(b_{ks}^{(1)}\right) \cdot \min\left(b_{ks}^{(2)}\right) \end{bmatrix}$$

Bin Creation and Merger

Bins are updated as new kernels enter the system. For the *bin list*, there are two cases to address when K_{new} enters: (1) the number of bins in *bin list* is less than Q; and (2) the number of existing bins is equal to Q. For both cases, if K_{new}'s center intersects any one of the bins, then a reference to K_{new} is added to the bin's kernel set, b_{ks}, and forwarded to the global *kernel set*. Suppose that the number of existing bins is less than Q, and K_{new} does not intersect any bin. Since the number of bins is less than Q, a new bin is created with K_{new} as its first and center point. The bin management algorithm continues in this manner until Q bins have formed. Now assume that there are Q bins in the *bin list* and K_{new} does not intersect any bin. In this scenario, a new bin is created for K_{new} and two bins (including one formed by K_{new}) are merged to maintain Q number of bins. The merger of two bins is defined by its additivity property (Equation 3.6). Because the bins represent local regions, they must remain continuous and mutually disjoint. Therefore, the merger can only occur between adjacent bins. The merger candidates are selected based on the objective function of the local region optimization (Equation 3.4). The following is an expanded expression of the uniformly weighted local region PAD variance (Equation 3.2) in terms of kernel centers $x_1, x_i, .., x_n$:

Equation 3.7

$$var(L_d) = \frac{2}{\|L\|-1}\left(\sum_{i=1}^{\|L\|} {x_i}^2 - \sum_{i=1}^{\|L\|-1} x_i x_{i+1} - \left(\frac{x_1^2 + x_{\|L\|}^2}{2}\right)\right) - \left(\frac{x_{\|L\|} - x_1}{\|L\|-1}\right)^2$$

By utilizing the above formula, the variance of the ith bin's pair-wise adjacent kernel distance is determined to be as follows (for uniformly weighted kernels):

Equation 3.8

$$\mathbb{B}_{dv}^{(i)} = \frac{2}{\left\|b_{ks}^{(i)}\right\|-1}\left(b_{ss}^{(i)} - b_{sp}^{(i)} - \left(\frac{\min\left(b_{ks}^{(i)}\right)^2 + \max\left(b_{ks}^{(i)}\right)^2}{2}\right) - \frac{(\max(b_{ks}^{(i)}) - \min(b_{ks}^{(i)}))^2}{2\left\|b_{ks}^{(i)}\right\|-2}\right)$$

Similarly, the ith bin's kernel weight variance is given as follows which is a reformulation of (Equation 3.3) in terms of the bin vector components:

Equation 3.9

$$\mathbb{B}_{wv}^{(i)} = \frac{b_{wss}^{(i)}}{\left\|b_{ks}^{(i)}\right\|} - \left(\frac{b_{ws}^{(i)}}{\left\|b_{ks}^{(i)}\right\|}\right)^2$$

Therefore, a merger will occur for the pair of adjacent bins which minimizes the following objective function:

Equation 3.10

$$\text{Min } J = \sum_{i=1}^{Q} \left\|b_{ks}^{(i)}\right\| (\mathbb{B}_{dv}^{(i)} + \mu \cdot \mathbb{B}_{wv}^{(i)})$$

$$\text{for } i = 1..Q \text{ and } \mu = 1$$

When K_{new} is added, updating all of the bin's features (with the exception of b_{sp}) follows straight-forward algebraic calculations and incurs constant time execution. To efficiently update b_{sp}, the following operations are employed:

1. Find the position of K_{new} within b_{ks} and set r to this index position. This implies that the kernels, $x_{r-1}$and x_{r+1}, are K_{new}'s adjacent neighbors
2. Update b_{sp} as follows:

Equation 3.11

$$b_{sp}{}^{(new)} = b_{sp}{}^{(old)} - x_{r+1}x_{r-1} + x_{r-1}x_r + x_r x_{r+1}$$

3.3.3 Kernel Maintenance

In order to maintain the kernels in a fixed memory environment, a kernel clustering paradigm is employed. If the size of the *kernel set* is M, then the insertion of K_{new} will cause an overflow and invoke the merger of two kernels at the corresponding bin. Let $G^{(1)}(x)$ and $G^{(2)}(x)$ be weighted kernels, then the merged kernel, $G^{(merge)}(x)$, is determined by utilizing a kernel merging approach as follows [46]:

Equation 3.12

$$G^{(1)}(x) = w^{(1)} K\left(\frac{x_1 - x}{h}\right)$$

Equation 3.13

$$G^{(2)}(x) = w^{(2)} K\left(\frac{x_2 - x}{h}\right)$$

Equation 3.14

$$G^{(merge)}(x) = (w^{(1)} + w^{(2)}) K\left(\frac{x_{merge} - x}{h}\right)$$

where x_{merge} is the merged kernel center of x_1 and x_2

$G^{(1)}(x)$ and $G^{(2)}(x)$ are selected such that the following L_2 error distance is minimized:

Equation 3.15

$$L_2 = \int_{-\infty}^{\infty} \left(G^{(1)}(x) + G^{(2)}(x) - G^{(merge)}(x)\right)^2 dx$$

It can be shown that the L_2 distance increases proportionally with the kernel distance, hence the mergers only occur between adjacent kernels. It can be shown that minimizing L_2 error can be done in constant time [41].

The remainder of this section presents the time-based sliding window algorithm which ensures that all elements in *kernel set* are within the time range, $[\tau_{min}, \tau_{max}]$. The algorithm is followed by a description of the density evaluation which leverages upon the *bin list* structure for efficient computation. Lastly, the selected kernel function and bandwidth form are provided.

Time-based Sliding Window

Let τ_{tl} be the length of the time window. To produce a sliding window, set τ_{max} to the current time and set $\tau_{min} = \tau_{max} - \tau_{tl}$. The *time list* is implemented as a First-In-First-Out queue with the head node being the oldest timestamp. When K_{new} is inserted into the *kernel set*, the time handling algorithm inserts the kernel's arrival time, g_{ta}, and the kernel's extended time span, g_{ets}, (i.e., time to remain in window *after* expiration and initially set to 0) to the tail of *time list*. Assume that two kernels, $G^{(1)}$ and $G^{(2)}$, are merged to become $G^{(merge)}$. The *time list* updating process proceeds as follows:

1. Remove the corresponding *time list* nodes of $G^{(1)}$and $G^{(2)}$
2. Define $G^{(merge)}$'s arrival time and time span as follows:

Equation 3.16

$$G_{ta}^{(merge)} = \min\left\{g_{ta}^{(1)}, g_{ta}^{(2)}\right\}$$

Equation 3.17

$$G_{ts}^{(merge)} = \max\left\{g_{ta}^{(1)} + g_{ets}^{(1)}, g_{ta}^{(2)} + g_{ets}^{(2)}\right\} - g_{ta}^{(merge)}$$

3. Insert $G^{(merge)}$'s arrival time into the time list

Kernel expiration is performed by comparing the *time list*'s head node, deleting all associated kernels with $g_{ta} < \tau_{min}$ and $g_{ets} = 0$ from the *kernel set*, and updating the *bin list*. As for expiring a kernel with $g_{ets} > 0$, assume that the weight of the kernel is uniformly distributed across its time span. Therefore, the kernel will remain in the *kernel set* but its weight will be adjusted to the following:

Equation 3.18

$$g_w^{(updated)} = \left(1 - \frac{\tau_{min} - g_{ta}^{(old)}}{g_{ets}^{(old)}}\right) g_w^{(old)}$$

where $g_w^{(i)}$ is the weight of kernel i.

3.3.4 Single Kernel Density Estimate Generation

The following describes the algorithm to solve a single density query which takes advantage of the local region pruning capability to provide high throughput performance. The corresponding kernel function and bandwidth forms are also discussed.

For a single query point d, the evaluation algorithm proceeds as follows:

1. **Bin search and kernel filtering:** Find the relevant set of bins which can potentially contribute a non-zero value to d. Let $h^{(\mathbb{B})}$ be the bandwidth of kernels in bin $\mathbb{B}$, then the bins can be found by scanning the *bin list* and selecting those which fulfill the following condition: $d \cap \left[\min(\mathbb{B}) - h^{(\mathbb{B})}, \max(\mathbb{B}) + h^{(\mathbb{B})}\right] \neq \emptyset$.
2. **Kernel search:** Scan the kernels within each relevant bin and sum the density contribution of kernels whose supports intersect d. Formally stated, let K be a kernel in $\mathbb{B}$, then compute the sum of kernel density contributions where $d \cap \left[K - h^{(\mathbb{B})}, K + h^{(\mathbb{B})}\right] \neq \emptyset$.

The evaluation technique presented above capitalizes upon the bin list structure by effectively pruning kernels which do not contribute to the final estimation result. Furthermore, the bandwidth is computed from the bin features in constant time.

Kernel Function and Bandwidth Form Selection

The class of admissible kernel functions must satisfy the conditions of Theorem 3.1 which requires that the kernels be (1) radially-symmetric and (2) compactly supported. The compact support property also eases the burden of computing the density estimates by eliminating kernels with $\left|\frac{x}{h}\right| \geq 1$. Some kernels of the admissible class are the Bi-weight, Rectangular, and Epanechnikov kernels [82]. It has been shown that the Epanechnikov kernel minimizes the asymptotic mean integrated squared error (AMISE) and

therefore it is optimal amongst all other kernels [56]. The Epanechnikov kernel is given as follows:

Equation 3.19

$$K(x) = \frac{3}{4}(1 - x^2) \text{ for } |x| < 1, \text{ and 0 otherwise}$$

Due to Epanechnikov kernel's compact support, radial-symmetry, and optimality w.r.t. the ASIME, this kernel is chosen for the proposed LR-KDE.

Each bin of the LR-KDE describes a region of similar density, hence, the captured distribution within each bin can be expected to be unimodal. Also recall that the LR-KDE assigns a unique bandwidth to each bin. Therefore, for each bin, the chosen bandwidth form is the *Scott's Rule* which has been shown to perform well under a wide range of unimodal distributions [82]:

Equation 3.20

$$h^{(\mathbb{B})} = \sqrt{5}\sigma^{(\mathbb{B})} \sqrt[-5]{n^{(\mathbb{B})}}$$

where $\sigma^{(\mathbb{B})}$ is the kernel centers' standard deviation of bin $\mathbb{B}$ and $n^{(\mathbb{B})}$ is the number of kernels in bin $\mathbb{B}$.

Note that $\sigma^{(\mathbb{B})}$ can be directly calculated from the bin features i.e., $\sigma^{(\mathbb{B})} = \sqrt{\frac{b_{wcss}}{b_{ws}} - (\frac{b_{wcs}}{b_{ws}})^2}$, therefore, computing the bandwidth, $h^{(\mathbb{B})}$, is achieved in $O(1)$ time.

3.4 Multiple Density Query Processing Framework

This section describes two main approaches to process multiple density queries within the LR-KDE. First, a multiple density query algorithm is provided that reduces the processing time of the single query approach by minimizing the redundant searches for local regions and kernel objects. Second, an optimized multiple density query algorithm is proposed that *approximates* the above exact multiple density query algorithm by utilizing constant time interpolation techniques. For the optimized approach, three variants are developed based on different query modeling approaches.

3.4.1 Multiple Kernel Density Estimate Generation

When evaluating multiple density queries using the single query algorithm (Section 4.4), the bin search and kernel filtering (Step 1) are executed in proportion to the number of query points. Additionally, the kernel search (Step 2) will visit non-contributing objects (if they exist) for each density query. Consequently, this results in redundant operations for queries that intersect identical set of bins. The situation occurs for any pair of queries that are strictly within $h^{(\mathbb{B}_1)} + h^{(\mathbb{B}_2)}$ distance apart where $\mathbb{B}_1 = \mathbb{B}_2$ or $\mathbb{B}_1$ and $\mathbb{B}_2$ are adjacent bins. Another source of inefficiency arises from the kernel search

routine where potentially non-contributing kernel objects are visited multiple times. Since these non-contributing kernels do not affect the final estimates, their processing should be minimized when evaluating multiple density queries. The set of non-contributing kernels are guaranteed to be empty for a query d when the intersecting bin's bandwidth-adjusted boundaries $\left[\min(b_{ks}) - h^{(\mathbb{B})}, \max(b_{ks}) + h^{(\mathbb{B})}\right]$ are strictly within the query's contribution range $\left(d - h^{(\mathbb{B})}, d + h^{(\mathbb{B})}\right)$. However, depending on the distribution of the query set, the queries can produce unnecessarily large amounts of visits to non-contributing kernel objects (e.g., queries close to the bin boundaries). The following multiple density query algorithms minimize the redundant processing associated with the single query algorithm.

Multiple Density Query Algorithm

The multiple density query algorithm for a given input query set $D = \{d_1, d_2, .., d_{|D|}\}$ is as follows:

1. **Query set preprocessing:** Sort the query elements into an array. Presorting the query elements will enable logarithmic searches of queries when processing the kernel objects.
2. **Kernel filtering:** Scan the *bin list* to obtain a set of bins for which their bandwidth adjusted intervals intersect the query range i.e., $[\min(D), \max(D)]$. This process removes and filters out kernel objects that are guaranteed to provide no contribution to D.
3. **Density aggregation:** For each $\mathbb{B}$ of the intersecting bins obtained from Step 2, visit each of the corresponding kernel object, perform a binary search on the sorted queries D using the kernel object center x as key, and calculate/aggregate the density contributions of all queries that intersect the kernel's bandwidth range $\left[x - h^{(\mathbb{B})}, x + h^{(\mathbb{B})}\right]$.

The multiple density query algorithm above removes the redundant filtering of kernels by invoking a one-time pass of the *bin list* to derive the filtered set of kernel objects. The redundant visits to non-contributing kernels within the filtered set is minimized by only invoking a single pass over the candidate kernels which can be much less than the multiple passes required (proportional to $|D|$) within the single density query algorithm. Lastly, the pre-sorting of the queries reduces each query search to a logarithmic cost when scanning the kernel objects.

3.4.2 Optimized Multiple Query Processing

The following describes the approach framework that further improves the multiple density query algorithm. If some errors with respect to the single density query are allowed, then the operations related to the kernel object search and aggregation can be reduced and supplanted by a fast interpolation technique. The following provides the optimized multiple density query approach.

Optimized Multiple Density Query Approach Framework

The optimized multiple density query approach for a given input query set $D = \{d_1, d_2, .., d_{|D|}\}$ is as

follows:

1. **Control point extraction:** Generate a sufficiently small set of control points from D (with size $s|D|$ where $0 < s \leq 1$) and output the control points in sorted order.
2. **Control point evaluation:** Estimate the densities of the control points obtained from Step 1 by employing the exact multiple density query algorithm.
3. **Query set interpolation:** Approximate the queries in D by utilizing a linear regression technique to interpolate density estimates. Other regression techniques can be used in this step to further enhance accuracy or reduce the number of control points required.

The above algorithm drastically reduces the calculation necessary for exact densities ($O\big((|D| + M)\log(|D|) + |D|M\big)$) and generates approximate solutions ($O(M\log(|C|) + |D|M)$ where $|C| \leq |D|$) through fast linear interpolations (see Section 6 for details on complexity analysis). In order to provide accurate results (relative to the exact method), the extracted control points are designed to model the distribution of the query set. Hence, three variants are proposed which employ differing approaches to generate the control points in Step 1. The three variants are random-based generation, uniform-based generation, and histogram-based generation.

The **random-based generation** technique employs a random sampling technique that produces a sorted set of independent and identically-distributed subsamples of the query set D which results in a total query cost of $O((|C| + M)\big(\log(|C|)\big) + |D|M)$ where $|C|$ is the size of the control points. In the **uniform-based generation** approach, the explicit requirement to sort the control points employed is removed. In this approach, equidistant and ordered control points for the span of D are generated that requires only a single pass on D. Hence, the resulting query cost is reduced to $O(M\log(|C|) + |D|M)$. However, a drawback of this approach is its explicit embedding of a uniformly distributed query set which can produce inaccurate control points for highly skewed query distributions. In order to better model the queries' distribution and avoid the need to sort the control points, the **histogram-based generation** is proposed. First, a histogram of the query set is calculated that will be used to guide the assignment of control points which can be achieved in a single pass of D. Second, the control points are assigned to locations that are locally equidistant within a histogram bucket and the ratio of locally equidistant points to the total number of control points is identical to the bucket's frequency. Using the histogram-based generation method, control points are generated that can effectively model varying query distributions (e.g., skewed distribution). Because the histogram based approach generates control points with a cost linear to $|D|$, the total query cost is $O(M\log(|C|) + |D|M)$. For all of the above three approaches, an initial pass over the query set is performed to determine the minimum and maximum elements which are required for interpolation.

3.5 Analysis

This section provides the consistency results and cost complexities of the LR-KDE. An important

criterion for any KDE is its consistency, that is, as the number of samples approaches infinity, the KDE converges to the true density. The time/space complexities of the online maintenance technique and density evaluation approach are analyzed to guarantee that the LR-KDE heeds the constraints of the data stream environment. Section 3.5.1 provides the proof of LR-KDE's asymptotic consistency. Section 3.5.2 analyzes the costs for maintaining the local regions and kernel objects within a sliding time window. Lastly, Sections 3.5.3 and 3.5.4 provide the density evaluation costs for the single and multiple density queries, respectively.

3.5.1 Asymptotic Consistency

Assume that all n samples are uniquely accessible and continuously persistent. To prove the consistency of the local region KDE, it suffices to show that the density estimate within a local region fulfills Parzen's condition for the asymptotic convergence of a KDE [69]. Parzen provides a sufficient condition described as follows:

If kernel $K(\cdot)$ is a bounded Borel function with

Equation 3.21

$$\int |K(t)|\, dt < \infty \ , \ \int K(t)\, dt = 1 \ \text{ and } \ |tK(t)| \to 0 \ \text{ as } \ |t| \to \infty$$

and bandwidth h_n indexed on n sample points satisfies

Equation 3.22

$$h_n \to 0 \ \text{ and } \ nh_n \to \infty \ \text{ as } \ n \to \infty,$$

then for the KDE, $\hat{f}(x)$, and true density, $f(x)$,

Equation 3.23

$$\hat{f}(x) \to f(x) \ \text{ in probability as } \ n \to \infty$$

Since a local region employs the Epanechnikov kernel, the kernel conditions given by Parzen are completely satisfied. Recall that the bandwidth selected for a local region is the Scott's Rule: $h_n = \sqrt{5}\sigma \ \sqrt[-5]{n}$. Hence, it can be seen that the following holds: $h_n \to 0$ as $n \to \infty$. Therefore, Parzen's first bandwidth condition is satisfied. As for the second condition, note that a local region is continuous and has compact support, which implies

Equation 3.24

$$\sup(\sigma) = c, \text{ where } c \text{ is a constant}$$

which implies

Equation 3.25

$$\frac{h_n}{n} = \frac{\sqrt{5}\sigma \sqrt[-5]{n}}{n} = \frac{\sqrt{5}c}{n^{6/5}} \to 0 \text{ as } n \to \infty \Longrightarrow nh \to \infty$$

Therefore,

Equation 3.26

$$\hat{f}(x) \to f(x) \text{ in probability as } n \to \infty$$

3.5.2 Online Maintenance Complexities

Let $InsertCost_{total}(K_{new})$ be the total cost of inserting K_{new}, then the total cost of the insertion procedure is:

Equation 3.27

$$InsertCost_{total}(K_{new}) = Insert_{binlist}(K_{new}) + Insert_{kernellist}(K_{new}) + Insert_{timelist}(K_{new})$$

Bin list cost: $Insert_{binlist}(K_{new})$ is composed of a sequential search over the *bin list* and merging a pair of bins. Hence the cost of inserting K_{new} into the *bin list* is

Equation 3.28

$$Insert_{binlist}(K_{new}) = O(Q)$$

Kernel set cost: Similar to the bin insertion cost, the insertion to the *kernel list* is dominated by the kernel search and the L_2 error updates for each pair of adjacent kernels within the affected bin. Let R be the number of kernels in the updated bin, then the total cost of the *kernel list* insertion is:

Equation 3.29

$$Insert_{kernellist}(K_{new}) = O(R)\ )$$

Time list cost: The dominant cost for inserting into the *time list* is the removal of all expired kernels. This operation involves updates on the L_2 errors and bin statistics. Let S be the number of expired kernels and T be the total number of kernels within a bin of an expired kernel, then the cost of inserting K_{new} into the *time list* is:

Equation 3.30

$$\begin{aligned} InsertCost_{timelist}(K_{new}) &= Remove_{kernellist}(K_{new}) + Update_{bin} + UpdateL_{2\,kernellist} \\ &= O(S) + O(Q) + O(T) = O(S) \end{aligned}$$

Since $Q, R, S \leq M$, where M is the maximum size of the *kernel list*, the total cost of inserting K_{new} is:

Equation 3.31

$$InsertCost_{total}(K_{new}) = O(M)$$

Total space cost: The total space cost of the online maintenance algorithm is derived from storing the three primary structures, *bin list*, *kernel list*, and *time list* in memory:

Equation 3.32

$$SpaceCost_{total} = O(Q) + O(M) + O(M) = O(M)$$

The above analysis shows that the time and space complexities of the proposed online kernel maintenance algorithm are $O(M)$. Since M is fixed, the proposed maintenance strategy provides constant runtime and space complexities which meet the linear-pass constraint.

3.5.3 Single Density Evaluation Complexities

A single density evaluation composes of a sequential search of the *bin list* and a scan of all kernels which provides a non-zero contribution to the query point. Let $EvalCost_{total}(D)$ be the total cost of determining the density at all the query points in D then the total evaluation cost is:

Equation 3.33

$$EvalCost_{total}(D) = \sum_{d \in D} \left(Search_{binlist}(d) + \sum_{q \in (binlist \cap d)} Search_{kernellist}(d, q) \right)$$
$$= \sum_{d \in D} (Q + M) = |D|Q + |D|Mp_{kernellist}$$

where $p_{kernellist}$ is the expected ratio of the kernel list that are visited for each query d.

Since $Q \ll M$ and $Q \perp M$, the evaluation runtime cost for a single query $d \in D$ is:

Equation 3.34

$$EvalCost_{single}(d) = Q + M = O(M)$$

In practical applications, only a fraction of the kernels contributes to d, which implies that $Search_{kernellist}(d, q) \ll M$. Therefore, the total evaluation cost can be much less than the asymptotic cost described above. The space cost of the density evaluation is $O(1)$ since the evaluation algorithm makes use of a single counter to store the current density sum. Similar to the time and space complexities of the maintenance algorithm, the evaluation runtime and space costs are bounded by $O(M)$ and hence meet the linear-pass constraint.

3.5.4 Multiple Density Evaluation Complexities

In this section, analyses of the exact multiple density query costs and the three variants of the optimized density query algorithms are provided. For the exact method, the computation of a density is composed of presorting the query set D, finding the set the of kernel objects that can potentially provide positive contribution to at least one element of D, and searching through the presorted query set. The total evaluation cost of the multiple density query algorithm for a query set D is:

Equation 3.35

$$
\begin{aligned}
MultiEvalCost_{total}(D) \\
&= Sort(D) + Search_{binlist}(D) \\
&+ \sum_{m \in (kernellist \cap D)} \left(Search_{queryset}(m) + Aggregate_{queryset}(m) \right) \\
&= |D|\log(|D|) + Q + M\log(|D|) + |D|Mp_{queryset} \\
&= (|D| + M)\big(\log(|D|)\big) + Q + |D|Mp_{queryset}
\end{aligned}
$$

where $p_{queryset}$ is the expected ratio of the query set D that are visited for each kernel object.

As shown above in Equation 3.35, the exact multiple density query approach reduces the *bin list* search to $(|D| + M)\big(\log(|D|)\big) + Q$ from $|D|Q$ of the single query algorithm (see Equation 3.33). Furthermore, due to the sorted query set D and utilization of a logarithmic search approach, visits to non-influenced query points are also diminished which leads to $p_{queryset} \leq p_{kernellist}$. The combined reduction results in a total cost reduction over multiple invocations of the single density query algorithm. The space cost of the multiple density query algorithm is $O(|D|)$ which is required to store the iteratively updated densities of the query set.

In the following, the evaluation cost of the three multiple density query variants are analyzed. For all the following cost analyses, C is defined to be the set of control points obtained from D and $|C| \leq |D|$. The first multiple density query algorithm variant, **random-based generation (RG)** approach, has the following evaluation cost:

Equation 3.36

$$
\begin{aligned}
RG_{MultiEvalCost_{total}}(D) &= (|C| + M)\big(\log(|C|)\big) + Q + |C|Mp_{queryset} + 2|D| - |C| \\
&= O\left((|C| + M)\big(\log(|C|)\big) + |D|M \right)
\end{aligned}
$$

The first three additive terms (i.e., $(|C| + M)\big(\log(|C|)\big) + Q + |C|Mp_{queryset}$) correspond to Step 2 of the algorithm where the exact multiple density query algorithm is invoked to obtain the density values of C. The latter two terms (i.e., $2|D| - |C|$) refers to the cost of determining the minimum/maximum elements in D and interpolating the remaining elements $D - C$. The space cost of the random-based generation approach is $O(|D|)$.

For the second multiple density query algorithm variant, **uniform-based generation (UG)**, the evaluation cost is as follows:

Equation 3.37

$$
\begin{aligned}
UG\_MultiEvalCost_{total}(D) &= M\log(|C|) + Q + |C|Mp_{queryset} + 2|D| \\
&= O(M\log(|C|) + |D|M)
\end{aligned}
$$

The uniform-based generation approach differs from the random-based generation cost by eliminating the

cost associated to query presorting ($|C|\log(|C|)$) and executing an extra $|C|$ number of interpolations. The space cost remains unchanged with $O(|D|)$.

The third multiple density query algorithm variant, **histogram-based generation (HG)**, has the following cost:

Equation 3.38

$$HG\_MultiEvalCost_{total}(D) = M\log(|C|) + Q + |C|Mp_{queryset} + 3|D| \\ = O(M\log(|C|) + |D|M)$$

The cost of the histogram-based generation approach imposes an additional operation over the uniform-based approach to perform the histogram estimation resulting in the last term $3|D|$. Asymptotically, the evaluation cost of the histogram-based generation is equivalent to the uniform-based approach. The space cost is $O(|D| + H)$ where H is the bucket size of the histogram.

3.6 Experiments

A set of comprehensive experiments have been conducted to validate the effectiveness and efficiency of the proposed online LR-KDE. The experiments focused on three core metrics: estimation quality, maintenance time, and density evaluation time. Other existing online KDE techniques were included for performance comparisons. The experiment design and metrics are given in Section 3.6.1. The results of the estimation quality of the LR-KDE and competing techniques are provided in Section 3.6.2. The single query LR-KDE density evaluation and construction costs are evaluated in Section 3.6.3. The multiple query LR-KDE density evaluation performance results and accuracy with respect to the single query method are described in Section 3.6.4. Lastly, a discussion of the experimental results is provided in Section 3.6.5.

3.6.1 Experiment Design

The experiments applied a battery of synthetic and real-world datasets to study the effects of various streaming conditions on the LR-KDE. For further details on the dataset, please see Appendix A. The datasets employed for this set of experiments were comprised of 2 synthetic (MIX2, MIX8) and 4 real-world (EEG, POWER, ROBOT, and TRAFFIC) time series data. The first 25K data samples were used for the experiments. The dataset reflects a wide range of streaming scenarios from simple (e.g., unimodal) to complex (e.g., multi-scaled and multimodal) distributions. The dataset also encompassed varying stationary properties with different periodicity (e.g., power demand load (POWER) and traffic volume (TRAFFIC)).

KDE Techniques and Parameters

Table 3.1 provides all of the evaluated techniques and parameters:

Name	Technique	Parameter
Sequence sample KDE	Sequence sample-based KDE [86, 91]	Max. # of kernels = 1000
Time sample KDE	Time sample-based KDE using the priority-sample algorithm [13]	Max. # of kernels = 1000
M-Kernel KDE	Variable bandwidth cluster KDE [98]	Max. # of kernels = 1000 Simplex max. iter. = 5000
Heinz KDE	List-based cluster KDE [46]	Max. # of kernels = 1000
LR-KDE	Single query Local Region KDE	Max. # of kernels = 1000, $\mu = 1,\ 4 \leq Q \leq 8$
Multi LR-KDE	Multiple query approach which gives identical estimates to the single query LR-KDE	Same as LR-KDE
Optimized Multi LR-KDE (SAMPLE)	Multiple query approach which employs a sampling based methodology	Same as LR-KDE, $s = .4$
Optimized Multi LR-KDE (UNIFORM)	Multiple query approach which employs a uniform gridding methodology	Same as LR-KDE, $s = .4$
Optimized Multi LR-KDE (HISTOGRAM)	Multiple query approach which employs a histogram methodology	Same as LR-KDE, $s = .4$

Table 3.1 Evaluated Online KDE Techniques

For all of the evaluated techniques in Table 3.1 (except for time sample KDE), the time windows were set to be the total length of the data stream.

Test Methodology

The first component of the experiments was to measure the estimation quality of the online KDE techniques. This was accomplished by establishing the ground truths for all datasets. In the synthetic case, the exact density structures are given. For the real-world datasets, the true densities are defined to be the density estimates produced by the offline AKDE using all available data samples. For the AKDE, the nearest neighbor KDE was used as the pilot estimate [82]. The error measure used was the Root Mean Square Error (RMSE) which is defined as follows:

Equation 3.39

$$RMSE(\hat{f}^{(\rho)}) = \sqrt{\frac{1}{1000}\sum_{i=1}^{1000}\left(f(x_i) - \hat{f}^{(\rho)}(x_i)\right)^2}$$

where $\hat{f}^{(\rho)}(\cdot)$ is the ρ density estimation technique and $x_1 .. x_{1000}$ are query points which uniformly divide the entire span of the distribution.

The maintenance time of an online KDE is defined as the *total* amount of time required to insert and process a given set of data points. This measures the efficiency of the online KDE in updating its kernel structures to match the current stream. The density evaluation time is defined as the *average* time to evaluate and compute a single density query. The density evaluation time was measured after all of the data points were processed. In these experiments, 10 trials were conducted for each evaluation component and the averaged results were reported.

Multiple Query Processing Evaluation

Experiments were also conducted on the multiple query approaches to test the impact on query efficiency and estimation quality. In particular, the cumulative query times were and the relative L_1 deviations to the single query algorithm were measured on various query sizes and profiles. The relative L_1 deviation is defined as follows:

Equation 3.40

$$RDEV(\hat{f}^{(LR_{SINGLE})}, \hat{f}^{(LR_{MULTI})}) = \frac{1}{1000}\sum_{i=1}^{1000}\left(\frac{ABS\left(\hat{f}^{(LR_{SINGLE})}(x_i) - \hat{f}^{(LR_{MULTI})}(x_i)\right)}{\hat{f}^{(LR_{SINGLE})}(x_i)}\right)$$

where $\hat{f}^{(\rho)}(\cdot)$ is the ρ density estimation technique and $x_1 .. x_{1000}$ are query points which uniformly divide the entire span of the distribution.

Query profiles were generated to reflect real-world mining applications which were categorized into two profiles: $\text{UNIFORM}_{\text{profile}}$ and $\text{SKEW}_{\text{profile}}$. $\text{UNIFORM}_{\text{profile}}$ reflects mining tasks that perform queries on a large domain interval of equidistant points. This type of query is employed in mining algorithms such as concept drift detection and visualization [41, 44, 45]. $\text{SKEW}_{\text{profile}}$ models queries that cover small subsets of the domain and tend to exhibit distributions that are skewed. This profile is modeled by a unimodal distribution with a randomly assigned center and scale. This query profile typifies mining tasks such as outlier detection and cluster analysis [48, 86].

Experimental platform

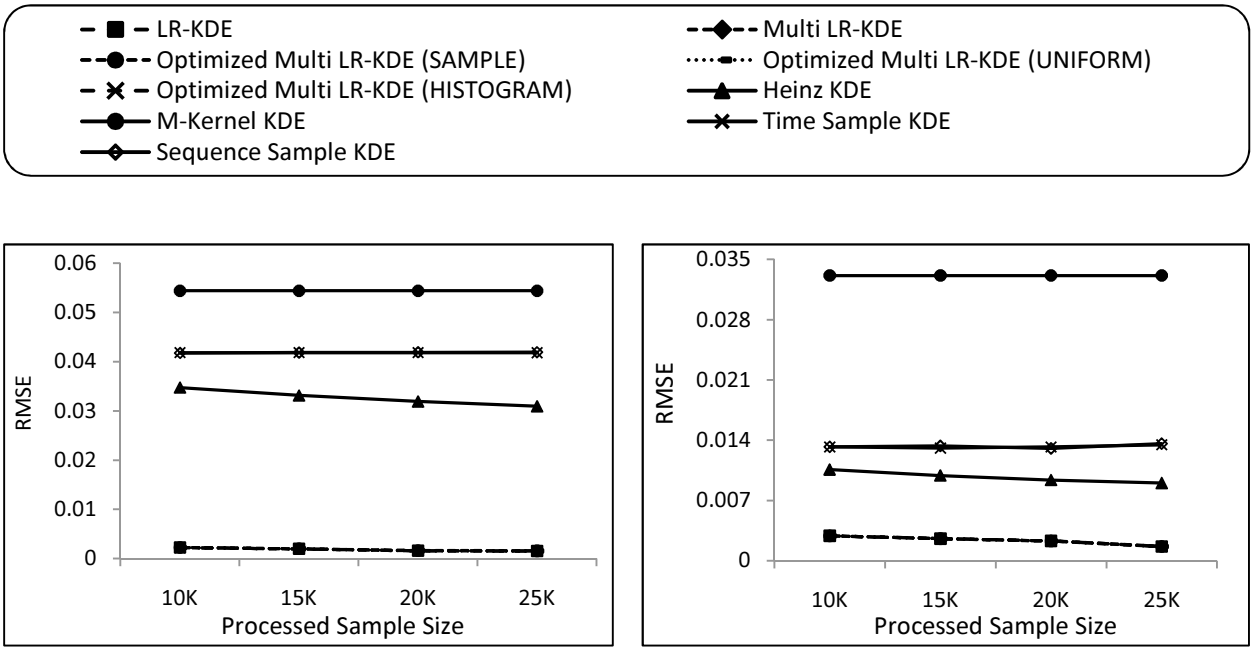
The experiments were conducted on a Windows Server 2003 Enterprise Edition (32-bit) operating system. The hardware platform was a 2.0 GHz Intel Pentium Core Duo 2 with 3 GB of RAM. This section is organized as follows.

3.6.2 Estimation Quality

For each metric and dataset, the LR-KDE was evaluated against the existing stream-based KDE techniques. The following provides the experiment results.

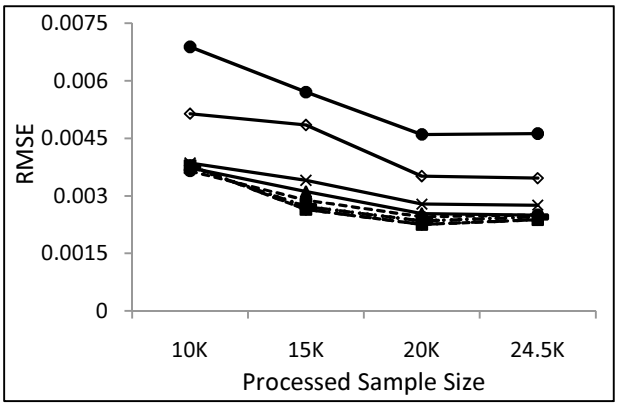
Figure 3.2 gives the estimation quality results of all datasets and KDE techniques. Each graph represents the estimation errors for a particular dataset where the *x-axis* is the number of processed sample points and the *y-axis* is the RMSE of the density estimates. The experiment showed that the LR-KDE (both single and multiple query approaches) provided lower RMSE than all competing techniques in the MIX2, MIX8, ROBOT, TRAFFIC, POWER, and EEG datasets. The LR based approaches produced significant RMSE reductions in MIX2 and MIX8 with errors that were at most half of the next best performing technique. Note that the time sample and sequence sample KDEs provided almost identical performance in MIX2 and MIX8.

Both the single and multiple query LR-KDE converged as more samples were processed in the MIX2, MIX8, POWER, and EEG datasets. However, for TRAFFIC and ROBOT, the RMSE of LR-KDE increased at 20K and 24.5K points, respectively. This behavior was similarly exhibited in other techniques such as M-Kernel KDE. In the TRAFFIC dataset, the sequence sample KDE produced a drastic change in its RMSE at the 20K mark. All of these observations suggest the presence of concept drifts in the POWER and EEG datasets.

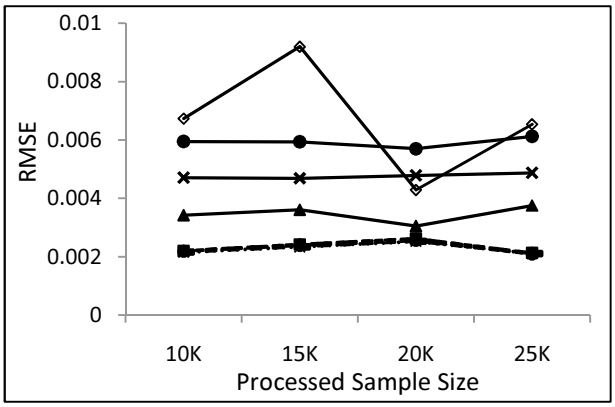


(a) RMSE of MIX2 dataset

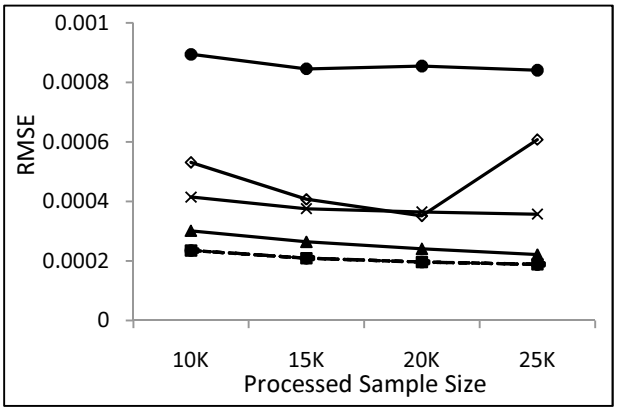
(b) RMSE of MIX8 dataset



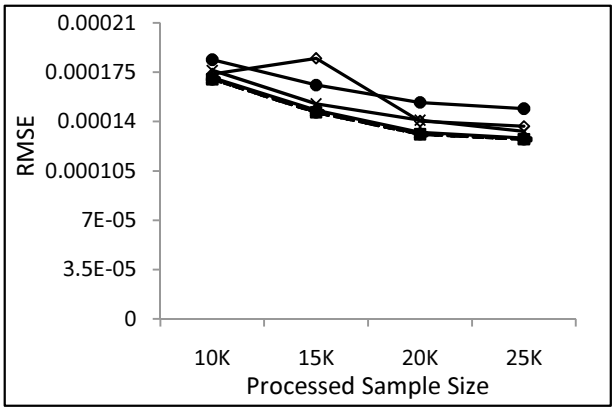
(c) RMSE of ROBOT dataset



(d) RMSE of TRAFFIC dataset



(e) RMSE of POWER dataset



(f) RMSE of EEG dataset

Figure 3.2 Estimation quality (RMSE) for all datasets

Figure 3.3 and Figure 3.4 show the plotted estimates of MIX2 and MIX8 by two of the lowest error attaining techniques, LR-KDE and Heinz KDE. The *x-axis* represents the query points and the *y-axis* shows the density. MIX2 and MIX8 possess multiple isolated modes which can be difficult to estimate with a single bandwidth KDE. For example, the Heinz KDE tended to oversmooth the distributions as indicated by the underestimated peaks and overestimated valleys. The oversmoothing can be attributed to Heinz KDE's use of the *Scott's Rule* bandwidth which is known to oversmooth multimodal distributions [82]. In an attempt to improve the accuracy of Heinz KDE, the only available parameter, kernel size, was increased from 1K to 100K. The increased kernel size produced ≤ 1% improvement in the RMSE and showed no observable difference in the plotted estimates. When the kernel size for LR-KDE was increased to 100K, it resulted in 5.5% (MIX2) and 7.8% (MIX8) improvements in the RMSE. Although the LR-KDE employs the *Scott's Rule* bandwidth, it restricts uniform bandwidth assignment to regions of similar densities. As a result, the LR-KDE identified all of the modes and accurately captured the critical distributional features.

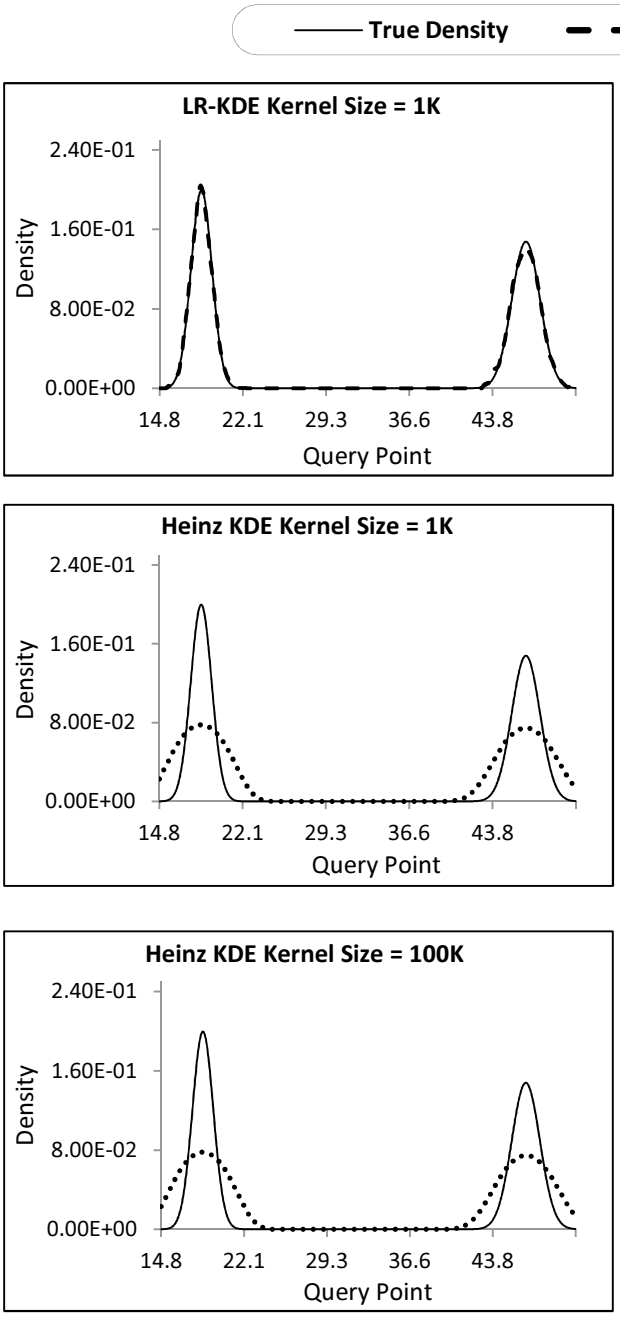


Figure 3.3 Plots of estimated densities by LR-KDE and Heinz KDE for MIX2

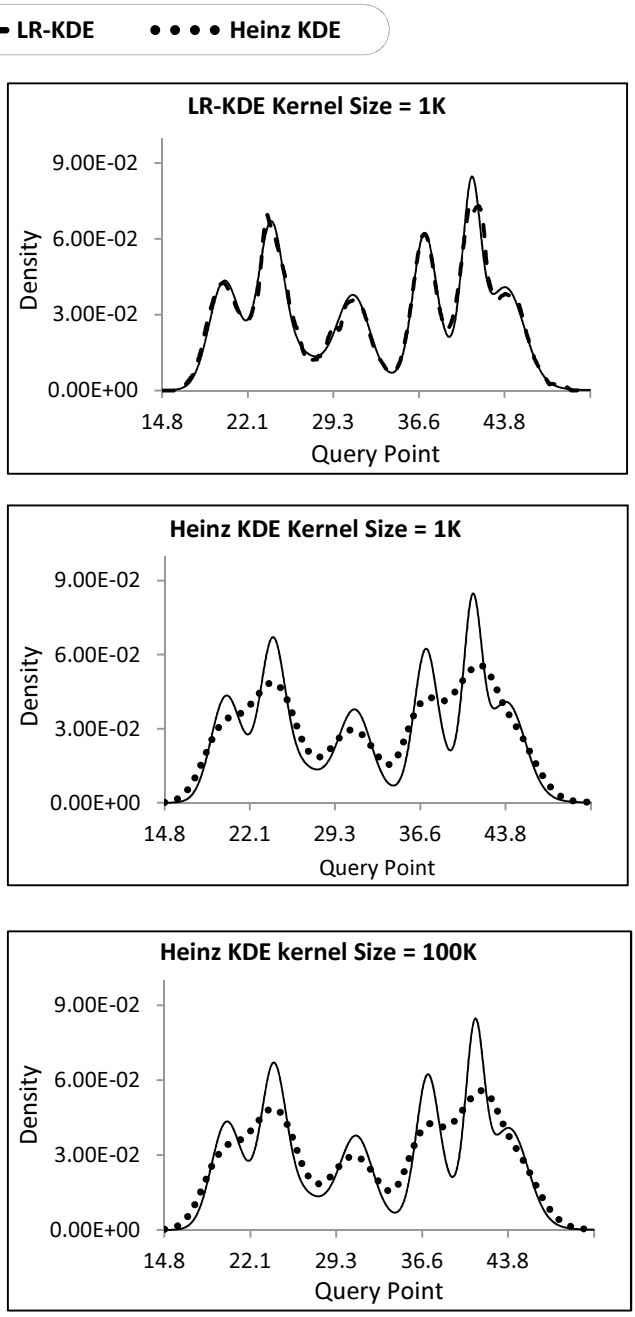


Figure 3.4 Plots of estimated densities by LR-KDE and Heinz KDE for MIX8

3.6.3 Single Query Processing

The following section shows the runtime results of updating the kernel objects and the evaluation times of the single query approaches.

Maintenance Time

Figure 3.5 illustrates the impact of the various data streams on LR-KDE's kernel maintenance times. The *x-axis* indicates the datasets and the *y-axis* measures the maintenance times for processing the entire data. The maintenance times of the multiple queries LR-KDE are identical to the single query approach since the only difference between the two methods is in their density query evaluation techniques. The real-world POWER, TRAFFIC, and ROBOT datasets showed the most improvements for the LR-KDE. This observation is attributable to the datasets' largely ordered samples which reduced the amount of scans the LR-KDE needed to perform. In the MIX8 and EEG datasets, the LR-KDE provided lower times than all of the non-sample-based approaches. Note that the sample-based techniques produced nearly identical results within the various datasets.

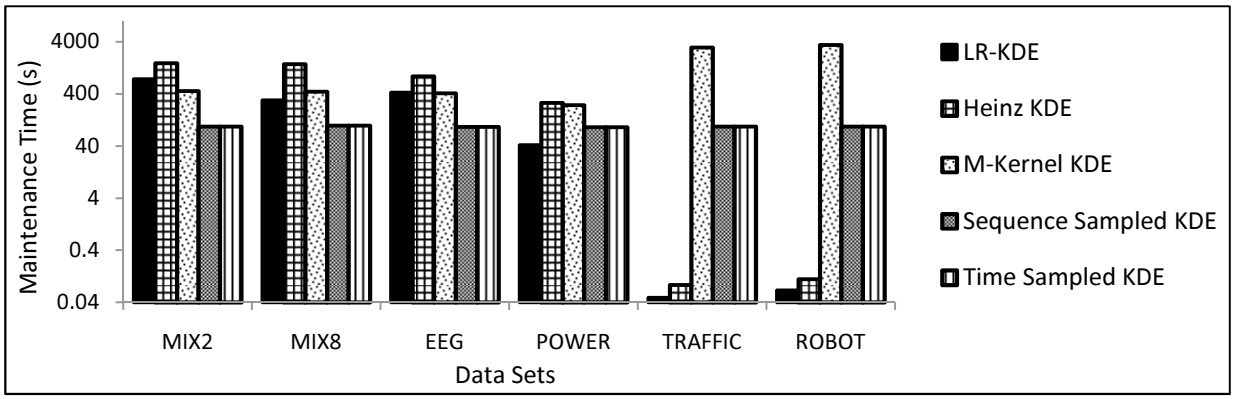


Figure 3.5 Log scaled maintenance time of all datasets

Figure 3.6 shows the relationship between maintenance time and sample size. The *x-axis* is the number of samples processed and the *y-axis* is the maintenance time. The POWER dataset is shown but similar trends can be observed in the other datasets. All of the KDE techniques exhibited times that were linear to the sample size; however, LR-KDE and Heinz KDE provided the lowest cost rates in POWER, TRAFFIC, and ROBOT. The linear trend of LR-KDE is also consistent with the analyses of Section 6. Standard deviation for all trials was ≤ 5%.

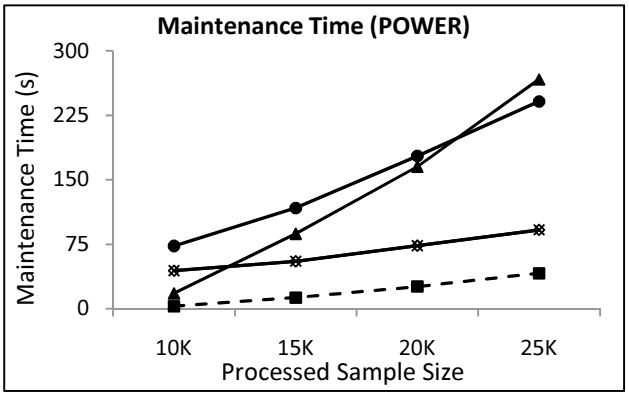


Figure 3.6 Maintenance times of POWER

Density Evaluation Time

Figure 3.7 shows the density query times of all datasets. The *x-axis* represents all of the datasets and the *y-axis* gives the average evaluation time for a single density query. For MIX2, MIX8, EEG, and POWER datasets, the LR-KDE consistently produced lower evaluation times than all of the competing techniques. For TRAFFIC and ROBOT, the LR-KDE performed equally to Heinz KDE but better than the other techniques. For the sample-based KDE, regardless of the kernel function employed, the entire kernel set must be scanned to generate a density estimate. This approach resulted in a consistent, but higher evaluation times than the LR-KDE and Heinz KDE. In summary, the results demonstrated that the LR-KDE evaluation performance was better than or at least equal to all of the competing techniques. Standard deviation for all trials was ≤ 10%.

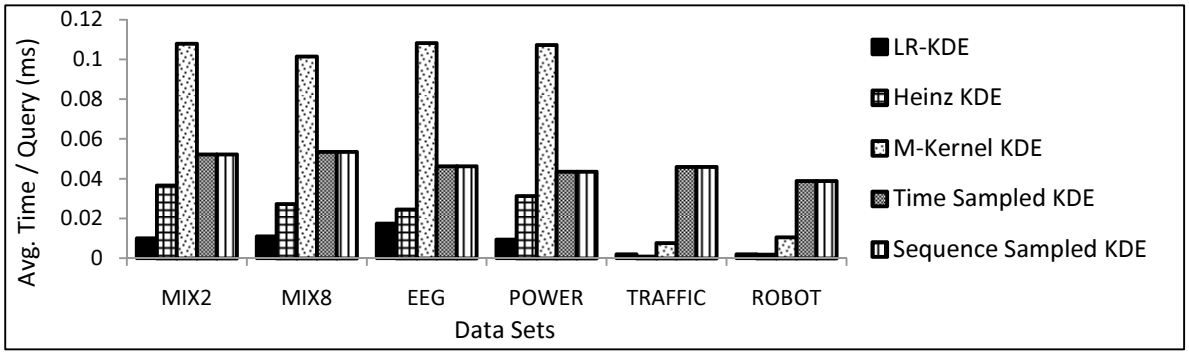


Figure 3.7 Average density evaluation time for a single query

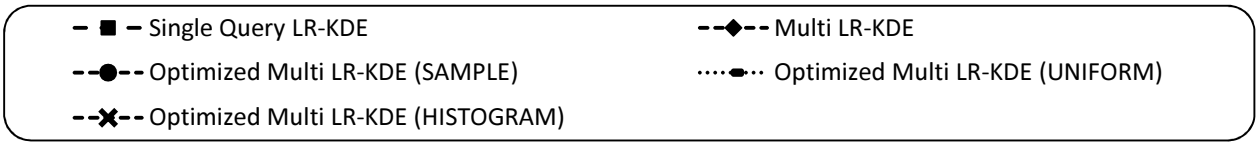
3.6.4 Multiple Query Processing

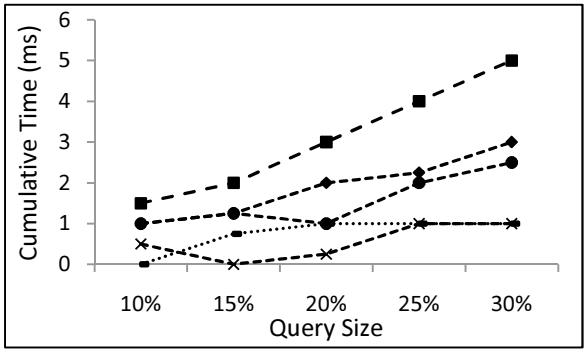
This section provides an in-depth evaluation of the multiple query approaches for the LR-KDE. Query

runtimes are measured for all datasets under varying query profiles and sizes. The relative deviations with respect to the single query approach are also measured for the optimized methods.

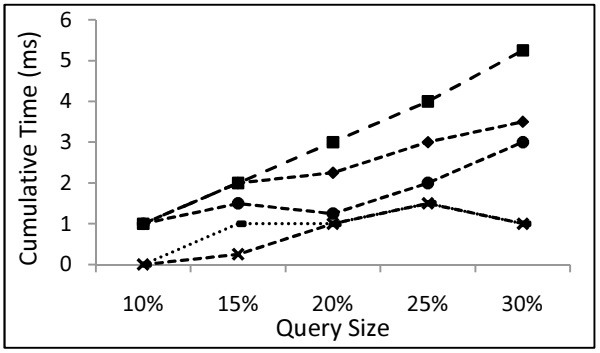
Density Evaluation Time

The query times of the single and multiple query LR-KDEs are provided in Figure 3.8 and Figure 3.9. The *x-axis* represents the query size ratio with respect to the kernel set size and the *y-axis* gives the average cumulative times. For all datasets and query profiles (i.e., $\text{UNIFORM}_{\text{profile}}$ and $\text{SKEW}_{\text{profile}}$), the multiple query approaches consistently match or outperform the single query method. Furthermore, the optimized techniques provide lower cumulative query times than the non-optimized multiple query approach. However, between the optimized techniques, the UNIFORM and HISTOGRAM techniques give faster running times than the SAMPLE based approach due to the elimination of query pre-sorting and linear time control point construction. In the ROBOT and TRAFFIC datasets, the single query LR-KDE gives very low query times (see Figure 7) which results in even faster times (near 0) by the multiple query approaches. Overall, the exact multiple query approach was effective in significantly lowering the evaluation of the single query approach for all the datasets and query profiles. These runtimes were further reduced by the optimized techniques which produced dramatic improvements over the single query approach.

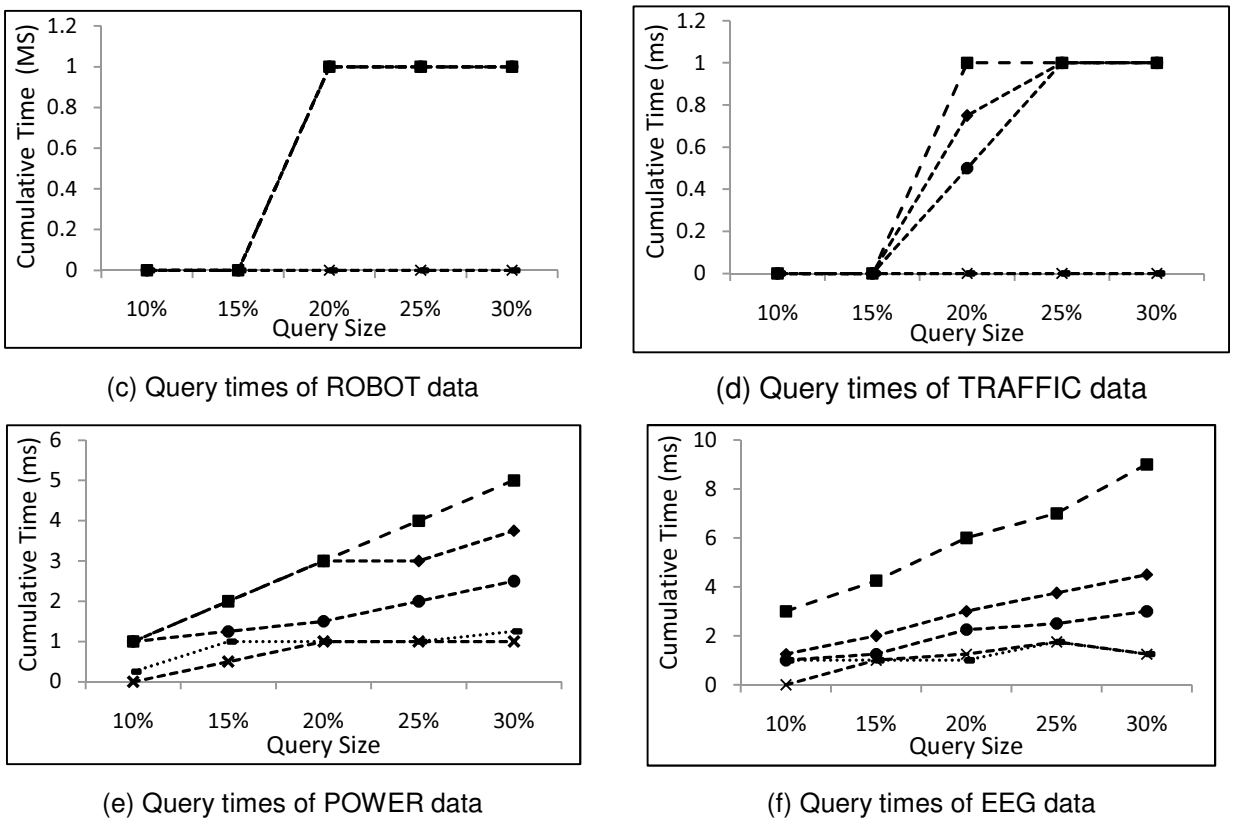




(a) Query times of MIX2 data



(b) Query times of MIX8 data



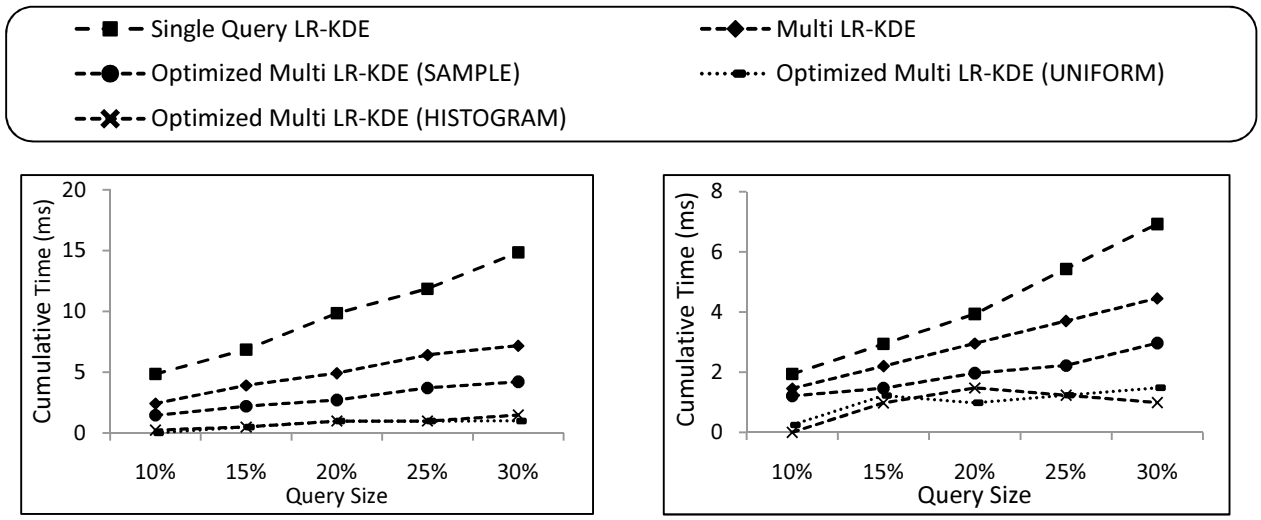
(c) Query times of ROBOT data

(d) Query times of TRAFFIC data

(e) Query times of POWER data

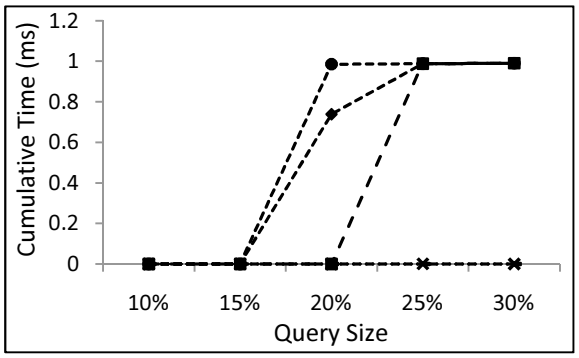
(f) Query times of EEG data

Figure 3.8 Query times for $\text{UNIFORM}_{\text{profile}}$

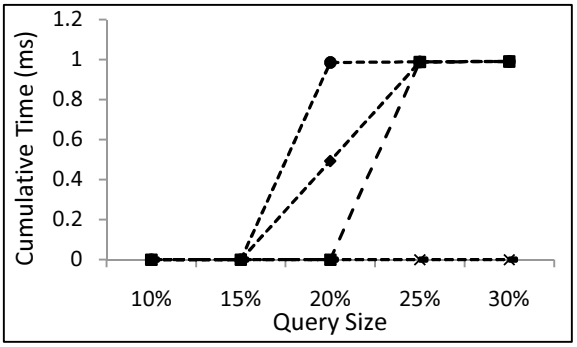


(a) Query times of MIX2 data

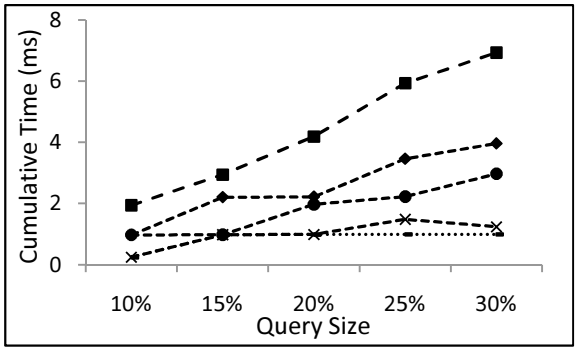
(b) Query times of MIX8 data



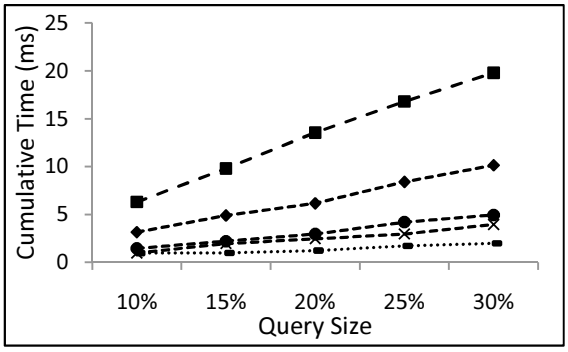
(c) Query times of ROBOT data



(d) Query times of TRAFFIC data



(e) Query times of POWER data



(f) Query times of EEG data

Figure 3.9 Query times for $SKEW_{profile}$

Relative Deviation

Figure 3.10 and Figure 3.11 show the mean relative L_1 deviation ratio to the single query algorithm for all query profiles. The results of the exact multiple query approach are not given since they generate identical estimates to the single query algorithm. In Figure 10 ($UNIFORM_{profile}$), all of the estimates produce less than 9% deviation while the HISTOGRAM and UNIFORM techniques both give less than 7.5% deviation. The HISTOGRAM and UNIFORM based variants consistently produced lower deviations than the SAMPLE based technique due to its ability to model sparsely populated query points. The SAMPLE based technique requires higher sampling rate to produce more accurate control points. However, for the $SKEW_{profile}$, the SAMPLE and HISTOGRAM based techniques are able to obtain significantly lower deviation than the UNIFORM based approach. In this scenario, the uniform assumption of the query distribution is not met which results in higher deviation for the UNIFORM approach. Because the HISTOGRAM and SAMPLE based techniques do not impose such a stringent assumption on the queries' distributional structure, these techniques produce significantly lower deviations.

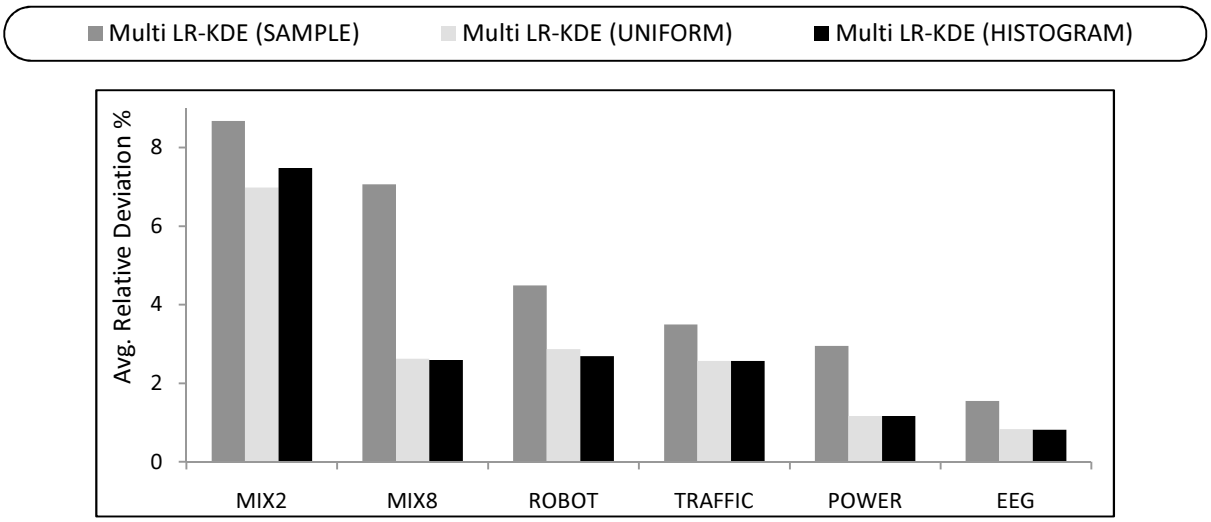


Figure 3.10 Relative deviation from single query algorithm for $UNIFORM_{profile}$

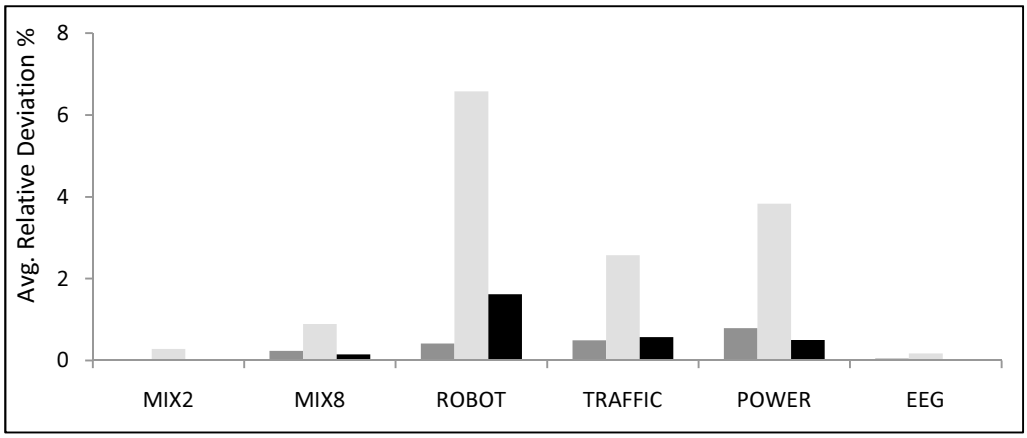


Figure 3.11 Relative deviation from single query algorithm for $SKEW_{profile}$

3.6.5 Discussion

Application of the local region concept to kernel density estimates has shown to be effective in modeling the local density features in data stream. As a result, the LR-KDE provided superior estimation quality over the competing techniques in both real-world and synthetic datasets. The LR-KDE was also able to improve the estimation accuracy in datasets which did not exhibit strong localities (e.g., predominantly unimodal datasets such as ROBOT). Since real-world data streams can exhibit strong local features (e.g., clustered outliers), it can be expected that stream mining applications would benefit from the use of the LR-KDE.

LR-KDE also improved the computational efficiency over the competing techniques. Local regions allow non-relevant kernels to be pruned from further processing which results in the simultaneous reduction of maintenance and query times. Furthermore, the LR-KDE retained the same order of space cost as the other online KDE techniques. The multiple query algorithms for LR-KDE can further improve the throughput of density estimates for mining tasks which generate multiple point queries. These cost reducing techniques are essential to stream mining tasks where results need to be furnished in real-time and processed in a fixed-size memory environment.

A concrete mining task that would benefit from LR-KDE's improved estimation quality and throughput is density-based clustering. In particular, those density-based approaches that employ bump hunting algorithms for determining the cluster centers [82]. For example, Figures 3 and 4 show that the LR-KDE captures and differentiates all of the modes almost exactly which would allow the bump hunting method to optimally isolate the cluster centers. In contrast, the next best performing technique (Heinz KDE) could not capture some of these modes which would increase the likelihood for the bump hunting algorithm to dismiss some potentially vital clusters. Such false dismissals can lead to missed emergent events and potential disastrous results. Within the context of a real-time environment, minimizing the time required to discover emergent events is essential to the overall success of the mining algorithm.

Density estimation can also be applied to the problem of outlier detection. An outlier can be defined as a sample point whose probability of occurrence falls below a predefined threshold [86]. Suppose that density estimates are employed to perform the above outlier detection scheme, then the rates of false dismissals are dependent on the accuracy of the estimated model. For instance, if the estimate is oversmoothed, then the rate of false dismissals may be increased in regions of low probability. In these regions, the true density of the sample points may be much smaller than their estimated values. Hence, a more accurate density estimation model, such as LR-KDE, can help reduce the false dismissals and improve the outlier detection performance.

3.7 Conclusion

This chapter addresses the issue of developing an efficient and asymptotically consistent online adaptive density estimation technique to meet the stringent constraints of the data stream environment. In that endeavor, we propose an online and local region based AKDE framework (LR-KDE) for univariate streams. The contributions of this chapter include: the first KDE approach that supports adaptive bandwidth and efficient multiple query processing over data streams, the development of the pair-wise distance criterion that effectively approximates the AKDE and guarantees asymptotic consistency, the design of linear-pass algorithms to maintain and compute kernel density estimates over a time-based sliding window, and the construction of a set of fast algorithms to accurately estimate multiple density queries. Theoretical analyses are provided to validate the asymptotic consistency and computational complexities of the LR-KDE. Experiments demonstrated that the LR-KDE enhanced estimation quality,

improved maintenance performance, and reduced density evaluation time over the existing techniques. The experiments showed that LR-KDE can improve the effectiveness of real-time stream mining tasks such as clustering and outlier detection.

Chapter 4. Theoretical Properties of Local Region based KDE

This chapter provides a theoretical treatment of several fundamental aspects of the local region (LR) based KDE. In Chapter 3, the complexities of the LR-KDE algorithm were provided and its consistency established through the application of Parzen's sufficiency condition. This chapter extends the theoretical study of the LR based KDE in an attempt to provide a better understanding of its key properties and its implications on existing stream-based KDEs. Analyses of both the point-wise and global accuracy of the LR based KDE are provided. These estimation aspects are expressed in terms of the bias, variance, mean squared error, and mean integrated squared error. The error forms are then used to establish the estimator's asymptotic consistency. In data streams, consistency can ensure that the expected error will be reduced 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. However, under certain distributions (i.e., those with large values of the aggregated curvature), it is shown that the LR based KDE can provide lower expected error than the standard KDE. Specifically, the local region based approach can reduce the expected error of distributions that are structurally complex (e.g., composites of multimodal and multi-scaled features).

This chapter also discusses the prerequisites of applying the local region technique to existing stream-based KDEs. The results show that its application can improve the estimation quality of existing stream-based KDEs. This chapter also establishes a criterion for selecting the local region number to minimize the expected error and describes a heuristic to solve it. Lastly, an alternative score function for local region construction is proposed that supports the use of currently available clustering approaches.

In summary, this chapter addresses the following research tasks:

1. **Point-wise accuracy:** Derive and evaluate the expected performance of the LR based KDE. In particular, analyze the estimator's point-wise accuracy.
2. **Global accuracy:** Expand the analyses above and derive the global accuracy.
3. **Comparative analyses:** Analyze the convergence rate against the standard KDE, determine the conditions under which the LR based KDE can produce lower estimation errors, and investigate the impact of concept drifts to estimation quality.
4. **Local region construction:** Formulate an appropriate local region number to minimize the expected error and investigate an alternative local region generation criterion.
5. **Application to existing KDEs:** Study the impact of applying LR on the estimation quality of existing stream-based KDEs.

The proposed research tasks above can be transformed into various analyses of the estimators' asymptotic mean integrated squared error (AMISE). Most current studies of the KDE employ the AMISE as the standard loss function. The AMISE gives the expected error performance of the KDE under the complete

domain space for large sample sizes (technically, as size approaches infinity). Since data streams are potentially unbounded, the AMISE provides a natural measure of the performance of stream-based KDEs. The AMISE's tractability also makes it an attractive choice for many analytical studies. However, the AMISE has certain drawbacks which arise from the quadratic error term which can disproportionately amplify certain error values. Nonetheless, the AMISE can still serve as a suitable loss function to a variety of applications [82].

This chapter is organized as follows: Section 4.1 introduces the theoretical preliminaries. Section 4.2 discusses the point-wise accuracy of the local region based KDE. Section 4.3 extends the point-wise error analyses and derives the global error on the complete domain space. Section 4.4 provides an in-depth comparison against the standard KDE. Section 4.5 explores some critical aspects on local region construction. Section 4.6 investigates the potential benefits of applying the local regions to existing KDE techniques. Lastly, Section 4.7 gives the conclusion.

4.1 Preliminaries

Theoretical preliminaries are provided which include definitions and assumptions and some important observations on the Scott's Rule bandwidth. The discussion on Scott's Rule helps highlight the deficiency of existing stream-based KDEs and its potential impact within the local region framework.

4.1.1 Scott's Rule Bandwidth

This section describes some important properties of the Scott's Rule bandwidth and its role in stream-based KDEs. The majority of stream-based KDEs employ the global bandwidth form (i.e., standard KDE) and in particular, they utilize a specific bandwidth function called the Scott's Rule [5, 46, 82, 86]. The Scott's Rule bandwidth assumes a Gaussian distribution reference and thus has a tendency to mask local features and over-smooth 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 [86]. Then the generated distributional model can be over-smoothed and lead some estimates within the sparse regions to be biased upwards [19]. This effect can cause the detection scheme to dismiss some true outliers. In a mission critical application such as military surveillance, a single false dismissal can be disastrous.

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

Equation 4.1

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

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 bi-normal density each with 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 over-smooth. However, if the density is a simple unimode, then σ_D can accurately describe the density structure and the Scott's Rule will provide good estimates. The problem of Scott's Rule (i.e., its inability to appropriately represent the complete density structure using a single measure) highlights the essential issue underlying all global bandwidth KDE techniques.

4.1.2 Definition and Assumptions

This section provides the definition of the local region and its key properties. The assumptions on the kernel function and bandwidth form are also given. Table 4.1 lists the notations used in the proceeding sections.

Symbol	Definition
L	A set of local regions
l	The set of samples within a local region
$\hat{f}_{LR}(\cdot)$	Local region based kernel density estimator of $f(\cdot)$
$\hat{f}_{KDE}(\cdot)$	Standard kernel density estimator of $f(\cdot)$
$K_h(\cdot)$	Kernel function with bandwidth h

Table 4.1 Notations and descriptions

Definition 4.1. Local Region (LR): Let D be an i.i.d. sample set and $p = [a, b]$ be a subinterval of D, then an *unrestricted* local region of D is $l = \{z \mid z \in D \wedge (a \geq z \geq b)\}$. 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 local regions of D is $L = \left\{ l_j \mid \cap_{i \neq j}^{k} \left((\Omega(l_i) .. \Theta(l_i)] \cap \left(\Omega(l_j) .. \Theta(l_j)\right] \right) = \emptyset \right\}$ and $z \in D \Longrightarrow \Omega(l_i) < z \leq \Theta(l_i)$ for exactly one $l_i \in L$. Hence, L is a set of local regions that disjointly partitions the sample set D. Due to this disjoint constraint, each $l_j \in L$ is called a *restricted* local region. Throughout this paper, a local region (without a qualifier) refers to the *restricted* local region as defined above. The local region defined in this chapter is a generalization of the local region in Chapter 3. Previously, a local region was assumed to employ the minimization of the relative density variance construction criterion. Here, the construction criterion is unspecified and is dependent on the assumptions of the local feature structures.

Definition 4.2. LR Center and Radius: Let l be a local region, 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 local regions L, the LR based KDE $\hat{f}_{LR}(x)$ of D is defined as follows:

Equation 4.2

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

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

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

Equation 4.3

$$K(t) \geq 0 \ , \ \int K(t)dt = 1 \ , \ \int tK(t)dt = 0 \ , \ \int t^2K(t)dt < \infty$$

Assumption on Bandwidth Form: Each bandwidth h_j is positive and follows Parzen's sufficiency conditions [69]:

Equation 4.4

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

As stated in Definition 4.1, local regions provide a total and disjoint partitioning of the data sample set D. The criterion by which to partition D (i.e., construct the local regions) is dependent on the bandwidth function assigned to each local region. Hence, the specification of the local region construction is achieved after establishing the desired condition of the local estimates. For example, if a single local region 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 local regions.

Significance of the theoretical framework: To evaluate a given local region 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 chapter derives some concrete forms of the estimation error (e.g., mean squared error) 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, for example, the conditions for which the LR based KDE can obtain enhanced estimates over the standard KDE.

4.2 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 mean squared error (MSE).

4.2.1 Bias and Variance

This section 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 trade-off [40]. 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 4.31 w.r.t. standard KDE) is applied to each local region, then $\frac{\text{variance}}{\text{bias}} \to |D|^{-\frac{2}{5}} \to 0$ as $|D| \to \infty$ since bias $\propto h_j^2$ and variance $\propto \left(|D|h_j\right)^{-1}$. This result also holds when the Scott's Rule is applied to each local region due to its AMISE optimality w.r.t. 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 local region l_j, then the bias of $\hat{f}_{LR}(x)$ is given as follows:

Equation 4.5

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

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:

Equation 4.6

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

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

Equation 4.7

$$E\left[\hat{f}_{LR}(x)\right] = E\left[\sum_{i=1}^{m} \frac{K_{h_1}(x-z_i)}{|D|} + \sum_{i=m+1}^{|D|} \frac{K_{h_2}(x-z_i)}{|D|}\right]$$

$$= \frac{m}{|D|} E\left[K_{h_1}(x - \boldsymbol{Z_1})\right] + \frac{|D|-m}{|D|} E\left[K_{h_2}(x - \boldsymbol{Z_2})\right]$$

$$= \frac{m}{|D|} E\left[\frac{1}{h_1} K\left(\frac{x-\boldsymbol{Z_1}}{h_1}\right)\right] + \frac{|D|-m}{|D|} E\left[\frac{1}{h_2} K\left(\frac{x-\boldsymbol{Z_2}}{h_2}\right)\right]$$

Let $\boldsymbol{Z} = \boldsymbol{Z_1} \cup \boldsymbol{Z_2}$ and apply the definition of expectation:

Equation 4.8

$$E\left[\hat{f}_{LR}(x)\right] \leq \frac{m}{|D|} E\left[\frac{1}{h_1} K\left(\frac{x-\boldsymbol{Z}}{h_1}\right)\right] + \frac{|D|-m}{|D|} E\left[\frac{1}{h_2} K\left(\frac{x-\boldsymbol{Z}}{h_2}\right)\right]$$

$$= \frac{m}{|D|} \int \frac{1}{h_1} K\left(\frac{x-z}{h_1}\right) f(z)dz + \frac{|D|-m}{|D|} \int \frac{1}{h_2} K\left(\frac{x-z}{h_2}\right) f(z)dz$$

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

Equation 4.9

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

Using the 2nd order Taylor expansion for $f(\cdot)$ and substituting the result into the above expression, the following LR based KDE bias is derived:

Equation 4.10

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

The additive composition of Equation 4.8 for $|L| = k$ is $\frac{m_1}{|D|} \int \frac{1}{h_1} K\left(\frac{x-z}{h_1}\right) f(z)dz + .. + \frac{m_k}{|D|} \int \frac{1}{h_2} K\left(\frac{x-z}{h_2}\right) f(z)dz$ where $\sum_{i=1}^{k} m_i = |D|$. Using this generalized form, the expression above becomes the final bias expression shown in Equation 4.5. □

Lemma 4.2. Variance of LR based KDE: Let $f(x)$ be the PDF of D and h_j be the bandwidth of local region set l_j, then the variance of $\hat{f}_{LR}(x)$ is given as follows:

Equation 4.11

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

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

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

Equation 4.12

$$VAR\left(\hat{f}_{LR}(x)\right) = VAR\left(\sum_{i=1}^{m} \frac{K_{h_1}(x-z_i)}{|D|} + \sum_{i=m+1}^{|D|} \frac{K_{h_2}(x-z_i)}{|D|}\right)$$

Using the decomposition and weighted sum properties of variance, the estimator's variance expression can be isolated into its additive components:

Equation 4.13

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

$$= \frac{m}{|D|^2} VAR\left(K_{h_1}(x-\boldsymbol{Z_1})\right) + \frac{|D|-m}{|D|^2} VAR\left(K_{h_2}(x-\boldsymbol{Z_2})\right)$$

Let $\boldsymbol{Z} = \boldsymbol{Z_1} \cup \boldsymbol{Z_2}$ then

Equation 4.14

$$VAR\left(\hat{f}_{LR}(x)\right) \le \frac{m}{|D|^2} VAR\left(K_{h_1}(x-\boldsymbol{Z})\right) + \frac{|D|-m}{|D|^2} VAR\left(K_{h_2}(x-\boldsymbol{Z})\right)$$

The first term is derived as follows:

Equation 4.15

$$\frac{m}{|D|^2} VAR\left(K_{h_1}(x-\boldsymbol{Z})\right) = \frac{m}{|D|^2}\left(E\left[K_{h_1}(x-\boldsymbol{Z})^2\right] - E\left[K_{h_1}(x-\boldsymbol{Z})\right]^2\right)$$

$$= \frac{m}{|D|^2 h_1} f(x) \int K(s)^2 ds + O\left(\frac{m}{|D|^2 h_1}\right)$$

Derivation of the second term proceeds similarly as above. Combining both terms, the total variance is as follows:

Equation 4.16

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

The additive composition of Equation 4.14 for $|L| = k$ is $\frac{m_1}{|D|^2} VAR\left(K_{h_1}(x-\boldsymbol{Z})\right) + .. + \frac{m_k}{|D|^2} VAR\left(K_{h_2}(x-\boldsymbol{Z})\right)$ where $\sum_{i=1}^{k} m_i = |D|$. Using this generalized form, the expression above becomes the final variance expression shown in Equation 4.11. □

4.2.2 Mean Squared Error

The mean squared error (MSE) of an estimator, $\hat{\theta}$, is given as follows:

Equation 4.17

$$MSE(\hat{\theta}) = E[(\hat{\theta} - \theta)^2]$$

After applying the properties of expectation, the MSE can be rewritten as follows:

Equation 4.18

$$MSE(\hat{\theta}) = BIAS(\hat{\theta})^2 + VAR(\hat{\theta})$$

Hence, the mean squared error can be obtained via the estimator's bias and variance components. The resulting MSE of the LR based KDE is given in Lemma 4.3. Using the MSE, Theorem 4.1 shows that the point-wise estimate is L_2 consistent [57]. 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 4.19 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 point-wise 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 local region set l_j, and select h_j such that it satisfies the conditions of Equation 4.4, then the MSE of $\hat{f}_{LR}(x)$ is given as follows:

Equation 4.19

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

Proof. Using the definition of MSE (Equation 4.18) and substituting the bias (Equation 4.5) and variance (Equation 4.11) terms into the MSE, results in the following $\hat{f}_{LR}(x)$ MSE:

Equation 4.20

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

Because each h_j fulfills Parzen's sufficiency conditions (Equation 4.4), the MSE in Equation 4.19 is obtained. □

Theorem 4.1. MSE-Consistency of LR based KDE: Given the conditions of the kernel function (Equation 4.3) and assumptions on the bandwidth (Equation 4.4), the LR based KDE is MSE (point-wise) consistent.

Proof. To prove MSE consistency, we show that the LR based KDE's MSE (Equation 4.19) 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 $|D| \to \infty$:

Equation 4.21

$$\sum_{j=1}^{|L|} \frac{|l_j| h_j^2}{|D|} \leq \sum_{j=1}^{|L|} \frac{\max |l_{1 \leq i \leq |L||} \cdot \max \left| h_{1 \leq i \leq |L|}^2 \right|}{|D|}$$

$$= \frac{|L| \cdot \max |l_{1 \leq i \leq |L||}|}{|D|} \cdot \max \left| h_{1 \leq i \leq |L|}^2 \right|$$

$$= k \cdot \max \left| h_{1 \leq i \leq |L|}^2 \right| \to 0 \text{ as } |D| \to \infty$$

and

Equation 4.22

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

From the kernel conditions we have:

Equation 4.23

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

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

4.3 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* point-wise error ($\int MSE\left(\hat{f}_{LR}(x)\right) dx$) in the complete domain space. In Theorem 4.2, it is shown that the cumulative point-wise error (MISE) of the LR based KDE converges to zero, and therefore the LR based KDE is L_2 consistent within the support of the density. Notice that the $\|f''\|_2^2$ term in Equation 4.24 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 4.4.2 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 point-wise 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 4.2, 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 local region set l_j, and select h_j such that it satisfies the conditions of Equation 4.4, then the AMISE of $\hat{f}_{LR}(x)$ is given as follows:

Equation 4.24

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

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

Equation 4.25

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

Substituting the MSE (Equation 4.19) into the MISE expression above gives the following LR based KDE MISE:

Equation 4.26

$$MISE\left(\hat{f}_{h(i)}(x)\right) = \left(\sum_{j=1}^{|L|} \frac{|l_j| h_j^2}{|D|}\right)^2 \frac{\upsilon_2(K)^2}{4} \|f''\|_2^2 + \left(\sum_{j=1}^{|L|} \frac{|l_j|}{|D|^2 h_i}\right) \|K\|_2^2 + O\left(\sum_{i=1}^{|L|} \frac{|l_j| h_j^2}{|D|}\right) + O\left(\sum_{i=1}^{|L|} \frac{|l_j|}{|D|^2 h_j}\right)$$

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 4.4), the AMISE of the LR based KDE simplifies to Equation 4.24. □

Theorem 4.2. MISE Consistency of LR based KDE: Given the conditions of the kernel function (Equation 4.3) and assumptions on the bandwidth (Equation 4.4), the LR based KDE is MISE (globally) consistent.

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

Equation 4.27

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

Furthermore, from the kernel conditions we have:

Equation 4.28

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

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 4.26), we have:

Equation 4.29

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

4.4 Comparative Analysis

This section gives a comprehensive comparison between the LR based KDE and the standard KDE. Section 4.4.1 derives the convergence rate of the LR based KDE and shows that the rate is no lower than standard KDE. Section 4.4.2 gives an extensive analysis of the conditions by which the LR based KDE can attain lower expected error than the standard KDE. Section 4.4.3 explores the estimation capabilities of the LR based KDE and standard KDE under concept drifts. In summary, the analyses show that the LR based KDE can provide enhanced estimates of complex and evolving distributions.

4.4.1 Convergence rate

This section derives the convergence rates of the LR based KDE and compares it against the standard KDE (Equation 2.1). Since the following discussion includes a comparative evaluation of the standard KDE, the AMISE of the standard KDE is provided:

Standard KDE AMISE: Let $f(x)$ be the probability density function of D and $\hat{f}_h(x)$ be the kernel density estimator of $f(x)$ with a global bandwidth h, then the MISE of $\hat{f}_h(x)$ is given as follows [39]:

Equation 4.30

$$AMISE\left(\hat{f}_h(x)\right) = h^4 \frac{\upsilon_2(K)^2}{4} \|f''\|_2^2 + \frac{1}{|D|h} \|K\|_2^2$$

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

Because the convergence rate of a KDE is dependent on the selected form of the bandwidth, we derive a compatible bandwidth that is based on the standard KDE AMISE optimal bandwidth. The following provides the LR based KDE compatible bandwidth:

Equation 4.31

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

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

Substituting the bandwidth above into Equation 4.24 gives the following bandwidth specific AMISE:

Equation 4.32

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

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

Equation 4.33

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

From Equation 4.32 and Equation 4.33, the convergence rates of the LR based KDE and the global bandwidth approaches are identical. This result is not surprising since the number of local regions 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 4.24 and determining h_j which produces zero gradient. The steps in attaining a closed-form solution to this problem are complex as it requires 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 lower than $O\left(|D|^{\frac{-4}{5}}\right)$. In comparison to the fully adaptive KDE, the AKDE provides a faster convergence rate with order $O\left(|D|^{\frac{-8}{9}}\right)$. However, as alluded in Chapter 3, the AKDE exhibits a costly quadratic time complexity which does not meet the data stream requirements.

4.4.2 Error Reduction

This section 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 over-smoothed estimates of complex structures. However, it can give accurate estimates of simple features (e.g., unimodal density) and is amenable to efficient implementations [80, 82]. If each local region is tasked to capture the simple features of the density, then applying the Scott's Rule bandwidth to each local region 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: Suppose that the Scott's Rule bandwidth is applied to the standard KDE and to each local region of the LR based KDE (as shown in Equation 4.31) and $|D| > 0$, then

Equation 4.34

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

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

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 4.1).

Proof. Define the AMISE difference as $Z\left(\hat{f}_{KDE}, \hat{f}_{LR}\right) = AMISE\left(\hat{f}_{KDE}(x)\right) - AMISE\left(\hat{f}_{LR}(x)\right)$ and apply the Scott's Rule (Equation 4.1) to each estimator to obtain the following:

Equation 4.35

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

$\Longrightarrow$

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

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 4.1.

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

Equation 4.36

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

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

Equation 4.37

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

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

The above theorem shows that if the parameters of the LR based KDE satisfies Equation 4.34, 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 4.34 demonstrates the relations between the curvature $\|f''\|_2^2$ (i.e., complexity of the PDF), the AMISE difference $Z\left(\hat{f}_{KDE}, \hat{f}_{LR}\right)$, and the parameters of the KDEs in a somewhat complicated manner. To simplify these relations, we investigate the conditions for which

Equation 4.38

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

Because $\sqrt{|D|^{\frac{6}{5}}\sigma_D^4} > 0$ and $\sum_{i=1}^{|L|} |l_j|^{\frac{3}{5}}\sigma_j^2 > 0$, we have

Equation 4.39

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

Furthermore,

Equation 4.40

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

$$|D|^{\frac{3}{5}} - \sum_{i=1}^{|L|} |l_j|^{\frac{3}{5}} \frac{\max\{\sigma_1, .., \sigma_j, .., \sigma_{|L|}\}^2}{\sigma_D^2}.$$

The lower bound $|D|^{\frac{3}{5}} - \sum_{i=1}^{|L|} |l_j|^{\frac{3}{5}} \frac{\max\{\sigma_1, .., \sigma_j, .., \sigma_{|L|}\}^2}{\sigma_D^2}$ is minimized when each $|l_j| = 1$. 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

Equation 4.41

$$|D|^{\frac{3}{5}} - \sum_{i=1}^{|L|} |l_j|^{\frac{3}{5}} \left(\frac{\max\{\sigma_1, .., \sigma_j, .., \sigma_{|L|}\}^2}{\sigma_D^2}\right) > 0 \text{ when } \frac{\max\{\sigma_1, .., \sigma_j, .., \sigma_{|L|}\}}{\sigma_D} < |D|^{\frac{-1}{5}}.$$

Therefore,

if $\frac{\max\{\sigma_1, .., \sigma_j, .., \sigma_{|L|}\}}{\sigma_D} < |D|^{\frac{-1}{5}}$, then $\left(|D|^{\frac{6}{5}}\sigma_D^4 - \left(\sum_{i=1}^{|L|} |l_j|^{\frac{3}{5}}\sigma_j^2\right)^2\right) > 0$ which implies:

Equation 4.42

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

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

Based on the above expression, the LR based KDE can generate lower estimation errors than the standard KDE when the structural make-up of the true density is sufficiently complex i.e., high $\|f''\|_2^2$ value. This observation can also be implied from Equation 4.34 where the dominance of $\|f''\|_2^2$ can lead to the dominance of $\left(|D|^{\frac{6}{5}}\sigma_D^4 - \left(\sum_{i=1}^{|L|} |l_j|^{\frac{3}{5}}\sigma_j^2\right)^2\right)\|f''\|_2^2$ over $\alpha_2(K)\left(\sum_{i=1}^{|L|} \frac{|l_j|^{\frac{6}{5}}}{\sigma_j} - \frac{|D|^{\frac{6}{5}}}{\sigma_D}\right)$ which implies that $AMISE\left(\hat{f}_{KDE}(x)\right) - AMISE\left(\hat{f}_{LR}(x)\right) > 0$. In short, the higher the complexity of distributional structure (i.e., multi-modal and multi-scaled), the more likely that the LR based KDE will generate lower AMISE than the standard KDE. Recall that the experiments conducted in Chapter 3 showed that the LR-KDE produced higher margins of improvements for complex densities. Hence, the results of the experiments are consistent with the aforementioned analysis.

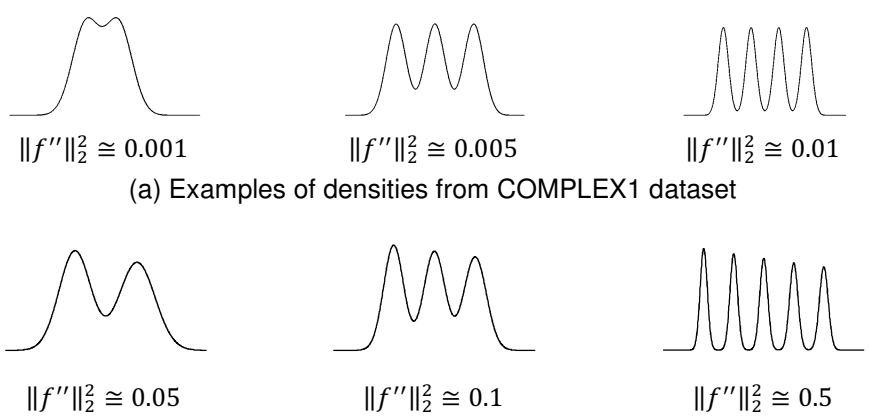


Figure 4.1 Generated mixture of normals with its aggregated curvature values from COMPLEX1 and COMPLEX2

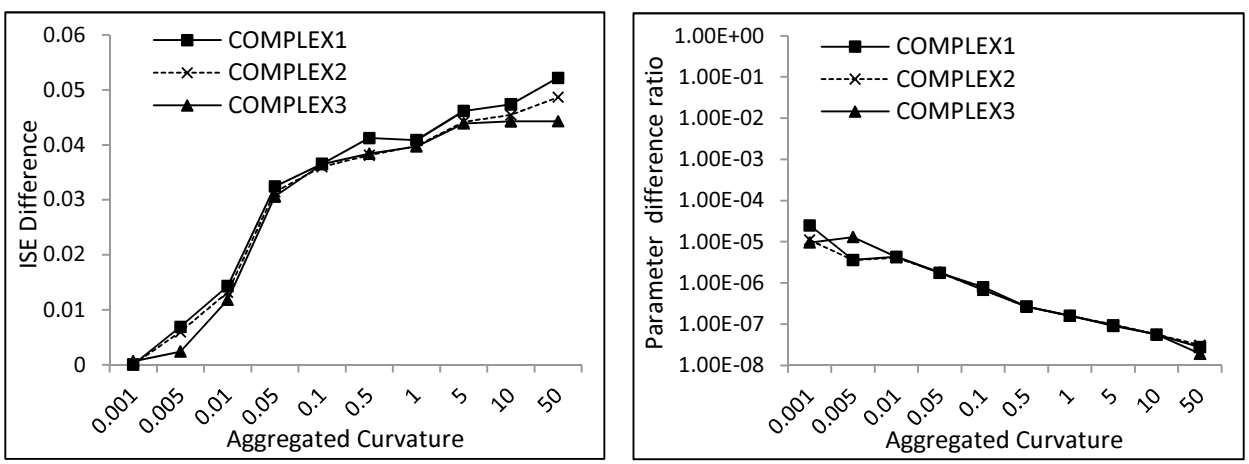
The following provides an empirical study of the relationship between the dataset's aggregated curvature $\|\boldsymbol{f}''\|_2^2$, the AMISE disparity between the standard KDE and LR based KDE, i.e., $\boldsymbol{Z}\left(\hat{\boldsymbol{f}}_{\boldsymbol{KDE}}, \hat{\boldsymbol{f}}_{\boldsymbol{LR}}\right)$, and the parameter difference ratio $\boldsymbol{\beta}(\boldsymbol{K}, \boldsymbol{L}, \boldsymbol{D})$. In particular, this experiment intends to demonstrate the effects of data complexity (as measured by its aggregated curvature $\|\boldsymbol{f}''\|_2^2$) 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 data were generated with varying degree of aggregated curvature $\|\boldsymbol{f}''\|_2^2$ values. The synthesized time series employed a mixture of normal density with the following canonical form: $\sum_{i=1}^{P} \boldsymbol{w_i N(c_i, s_i)}$ where $\boldsymbol{P}$ is the maximum number of modes. The first dataset, COMPLEX1, is composed of 10 time series with varying mode centers ($\boldsymbol{c_i}$) and fixed scales ($\boldsymbol{s_i}$) and weights ($\boldsymbol{w_i}$). The second dataset, COMPLEX2, is composed of 10 time series with varying centers and scales but fixed weights. The third dataset, COMPLEX3, is composed of 10 time series with varying centers, scales, and weights. Please see Appendix A for detailed descriptions of the dataset. Figure 4.1 shows some examples of the densities from COMPLEX1 and COMPLEX2 datasets. For each time series, its density was estimated using the standard KDE and the LR based KDE. This simulation utilized the ISE measure to approximate the AMISE difference $Z(\hat{f}_{KDE}, \hat{f}_{LR})$. The ISE difference is defined as $ISE(\hat{f}_{KDE}) - ISE(\hat{f}_{LR})$ where $ISE(\hat{f})$ is:

Equation 4.43

$$ISE(\hat{f}) = \sum_{i=1}^{1000} \Delta\left(f(x_i) - \hat{f}(x_i)\right)^2$$

where $x_1 .. x_T$ are ordered query points which uniformly divide the entire span of the distribution and $\Delta = \frac{x_T - x_1}{1000}$



(a) Aggregated Curvature vs. ISE Difference

(b) Aggregated Curvature vs. Parameter Difference Ratio

Figure 4.2 Plot of the relationship between aggregated curvature ($\|f''\|_2^2$), ISE difference ($\approx Z(\hat{f}_{KDE}, \hat{f}_{LR})$), and parameter difference ratio ($\beta(K, L, D)$)

Figure 4.2 provides the results of the simulation which 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., LR based KDE gave *lower* error than the standard KDE). This relationship is consistent with Theorem 4.3 and the analysis of Equation 4.42. Furthermore, the figure shows that when the data increases 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 in the LR based KDE over the standard KDE as the data's PDF becomes more complex (i.e., higher $\|f''\|_2^2$). From the results above, the LR based KDE shows that it can provide substantial estimation accuracy over the standard KDE for complex data streams.

4.4.3 Estimating under Concept Drift

An important result of previous section is the implications 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 bi-normal density which exhibits changes to its mode center distance (see Figure 4.3). Mode center variation can occur in data stream environment such as highway traffic. For example, roadway constructions can in many times shift the mode centers of normal traffic patterns. According to the parameter difference ratio $\beta(K, L, D)$ in Equation 4.42, 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 increase 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 4.42 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 the standard KDE. This adaptive characteristic of the LR based KDE is critical as it allows for high quality estimates under dynamic stream environments.

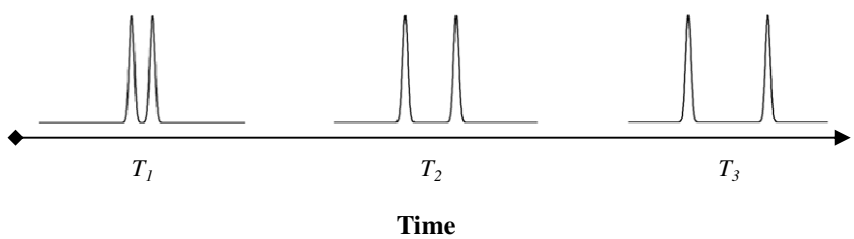
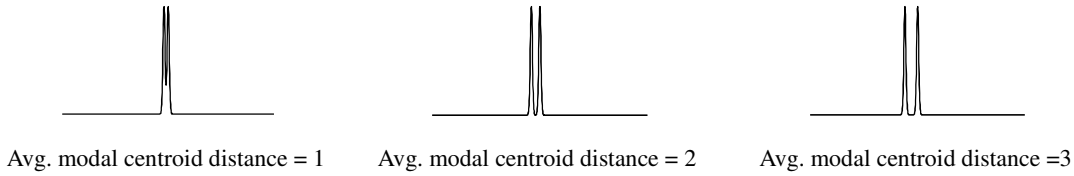


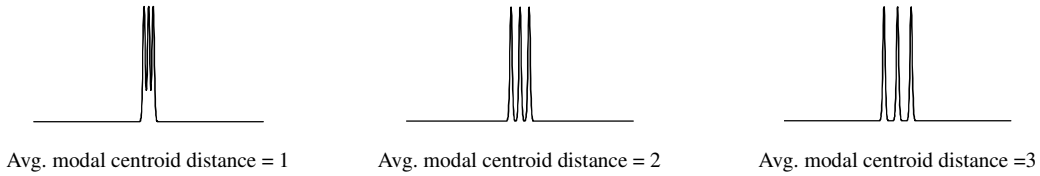
Figure 4.3 An example of concept drift due to modal distance shifts

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 with evolving mode center locations. Specifically, three datasets were produced which exhibit variation in mode center distances and mode counts. The first dataset, DRIFT1, contained 10 time series each with two modes in its PDF. The second dataset, DRIFT2, comprised of 10 time series each with three modes in its PDF. The third dataset, DRIFT3, composed of 10 time series data

with four modes. For each dataset, the average distance between the mode centers were varied at regular increments. Please see Appendix A for details on the datasets. Figure 4.4 gives some examples of the PDFs in DRIFT1 and DRIFT2 datasets.

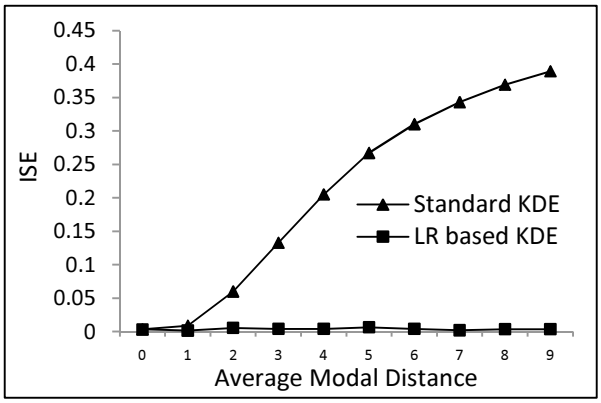


(a) Examples of densities from DRIFT1

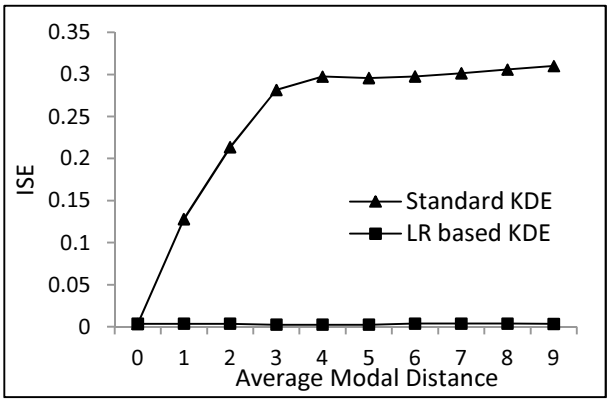


(b) Examples of densities from DRIFT2

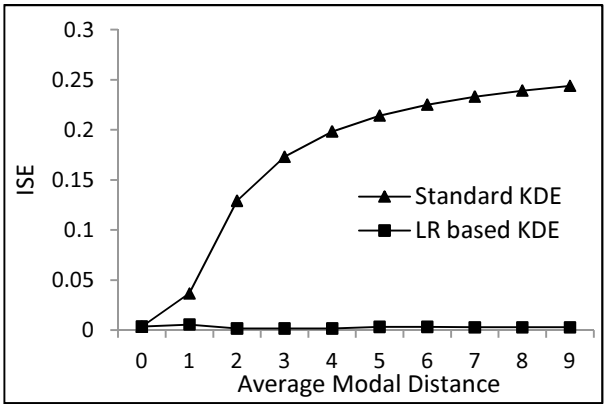
Figure 4.4. Examples of densities concept drift datasets DRIFT1 and DRIFT2



(a) Estimation quality of DRIFT1



(b) Estimation quality of DRIFT2



(c) Estimation quality of DRIFT3

Figure 4.5 KDE performance under modal shifts

Figure 4.5 shows the estimation results of the concept drift simulation. In all the datasets, the standard KDE error increases with the mode center distance while the LR based KDE generally remains the same. These behaviors of the standard KDE and LR based KDE directly support the analyses above. 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 since the increase in modal distance does not affect σ_j and $|l_j|$ of the parameter difference ratio. This observation coincide with the above analytical results where the estimation error of the LR based KDE was expected to be unaffected by the concept drift. The analyses and simulation results above show that under modal shifts, the LR based KDE can be employed to effectively mitigate the effects of changing density structures.

4.5 Local Region Construction

The construction of local regions involves two main steps: (1) determination of an appropriate local region number and (2) formulation of an appropriate construction criterion. The local region number in Step 1 loosely describes the relationship between the local regions (i.e., inter-region). The construction criterion of Step 2 expresses the desired condition of the intra-region state. Although there is a strong dependency between the local region number and the local region construction criterion, this section will study each component individually. The relationship between the two parameters will be discussed in Chapter 5 where a local region generation approach employing both parameters is proposed.

4.5.1 Local Region Number

This section discusses an appropriate local region number that minimizes the AMISE measure under the Scott's Rule bandwidth. Determining an appropriate local region number is a difficult task since it requires minimization of the AMISE which is a function of the unknown aggregated curvature $\|f''\|_2^2$. Depending on the characteristics of the aggregated curvature, various minimization policies can be utilized. For example, if it is assumed that the aggregated curvature and its multiplicative term generate a significant portion of the total AMISE, then a local region construction strategy targeted at reducing this product term can be employed. In cases where the density is structurally complex thereby producing a large aggregated curvature value, such a scheme would be able to produce improved estimation error than the standard KDE. If no specific assumptions on the density properties are made, then cross validation can be used to find an error minimizing local region number. However, care must be taken when employing such a technique since it generally assumes a static distribution and can involve high computational costs.

Since a PDF is composed of several modes with varying weights and scales, a reasonable approximation to the number of local regions is the true number of modes within the density function. This choice of a local number will allow the generation of sufficiently low value of the parametric difference ratio and thereby improving the opportunity of reducing the estimation error (see Section 4.4.2). The following provides some possible approaches for deriving the number of modes within a distribution. A mode can be regarded as a clustering of sample points. Hence, it is possible to apply clustering techniques to determine the number of modes within a distribution. One such technique is DenStream [20] which captures density-based clusters over an evolving data stream. DenStream employs the framework of CluStream [7] and utilizes the concept of micro-clusters. To determine the number of clusters, DenStream invokes the DBSCAN [27] algorithm on its maintained set of micro-clusters and outputs the number of resultant groupings. Other algorithms exists which purely aim at establishing the number of modes. Most of these algorithms are based on hypothesis testing such as one proposed by Cox which employs a series of evaluations on the data's histogram [23]. Another algorithm was developed by Silverman which employs the classical KDE estimated under varying bandwidths. Different bandwidth values are used to ascertain a critical value which allows the determination of multimodality conditions.

The following proposes an agglomerative strategy for estimating the number of modes within the data stream. The steps are outlined as follows:

1. Initially set the number of local regions to K_{max}.
2. Aggregate each nearest neighbor local region pair into a single local region in order of their distance.
3. Continue to aggregate each local pair until one of the following condition are met (see Section 5.2.1):
    a. Each aggregated local region is unimodal.
    b. Further aggregation of nearest neighbor local regions violates the unimodal condition.

The agglomerative scheme will require some further optimizations in the most computational intensive procedure: step 2. Because each aggregation in Step 2 requires a search of the nearest local region pair with the minimum distance, an index can be utilized to reduce the time to perform the search (see Sections 5.2.4 and 5.3.5).

4.5.2 An Alternative Local Region Construction Criterion

This section introduces an alternative criterion for generating local regions. The original objective function proposed in Chapter 3 provides an effective measure for generating locally and uniformly distributed partitions of the density domain space. Because the partitioning strategy is based on the pair-wise adjacent distance (PAD) theorem (Theorem 3.1), PAD imposes no restrictions on the choice of the bandwidth function. However, there exists a large class of KDE solutions that employ the Scott's Rule bandwidth [5, 41, 46, 80, 82, 86]. As a result, this section establishes an alternative local region criterion that is tailored to this bandwidth form. An important benefit of the new criterion is that it allows the use of a wide range of already available tools.

In deriving an alternative local region criterion, the LR based KDE AMISE under Scott's Rule is provided as follows:

Equation 4.44

$$AMISE\left(\hat{f}_{LR}(x)\right) = A\left(\frac{\sigma_1^4 l_1^{\frac{6}{5}} + \sigma_1^2\sigma_2^2 l_1^{\frac{3}{5}} l_2^{\frac{3}{5}} + \sigma_2^4 l_2^{\frac{6}{5}}}{|D|^{\frac{6}{5}}}\right) + B\left(\frac{l_1^{\frac{6}{5}}}{\sigma_1|D|^{\frac{6}{5}}} + \frac{l_2^{\frac{6}{5}}}{\sigma_2|D|^{\frac{6}{5}}}\right)$$

where $L = 2$, $A = \frac{\|f''\|_2^2 u_2(K)^2}{4|D|^{\frac{4}{5}}}$, $B = \frac{\|K\|_2^2}{\sqrt{5}|D|^{\frac{4}{5}}}$, σ_D is the standard deviation of D, and σ_k is the standard deviation of l_k.

Observe that Equation 4.44 involves estimations of the mode scales and local region sizes $|l_i|$. If each local region is assumed to model a unique mode as discussed in Section 4.5, then the new local region objective function are free to employ approximations such one provided by the EM Gaussian mixture algorithm. The EM algorithm's objective is to maximize the likelihood of the data fitting a mixture of Gaussian components. A variant of the EM algorithm, K-Means, employs the minimization of the sum squared error (SSE) criterion and assumes that the components have zero covariances (i.e., spherically shaped) and uniform weights. There are several works that have addressed the issue of developing efficient K-Means clustering approaches over data streams [7, 14, 36, 81]. Employing the SSE criterion to guide the local region construction will allow exploitation of this large and established set of tools.

To demonstrate the benefit of using the SSE criterion, a simulation study was conducted to compare the estimation qualities of the PAD and SSE criterions. The dataset is derived from MIX8.

Figure 4.6 and Figure 4.7 show the results of the two objective functions. The estimation results show that the SSE criterion can produce similar estimation quality to the PAD objective function. Hence, the SSE criterion could serve as a potential replacement to the PAD criterion. Furthermore, employing the SSE criterion will allow the local region discovery process to be implemented by existing techniques including those that handle multivariate data.

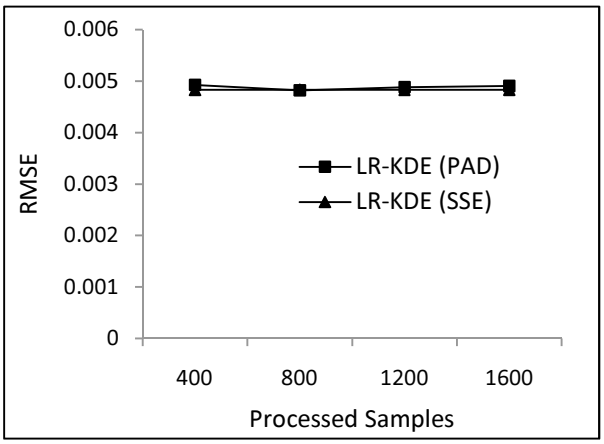


Figure 4.6 Estimation quality of PAD and SSE for various stream sizes with local region number = 6

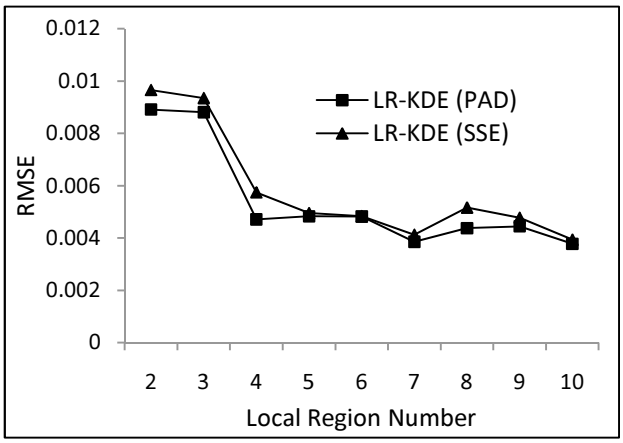


Figure 4.7 Estimation quality of PAD and SSE for various local region numbers for 1600 points

4.6 Application to Existing Kernel Representation Techniques

This section provides a discussion on the data sample properties as it is employed by the LR based KDE and its implications to existing stream-based KDEs. It can be seen from Chapter 3 that the LR based KDE can provide improved estimation quality over the global bandwidth KDE under multimodal distributions and mode shifts. The conditions for this result are embedded within the assumption of the AMISE function:

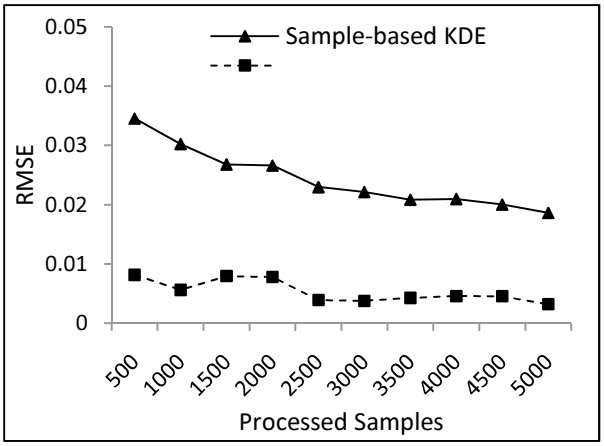
1. The samples of D are independently and identically distributed (i.i.d.).
2. The Scott's Rule bandwidth is employed.

No further discussion of the second condition will be made since the justifications for its use have been considered in previous sections and Chapter 3. Hence, the main thrust of the discussion will compose of the connection between i.i.d. samples and existing data representation approaches. Furthermore, the implication of this connection to current KDE techniques is discussed.

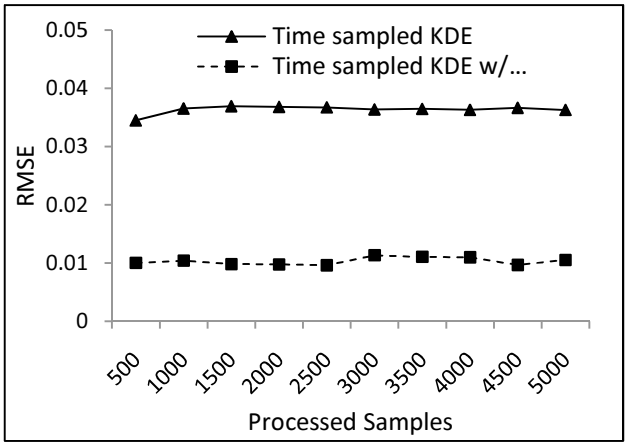
Data can be synopsized under various forms such as clustered objects, discretized grids, and sub-sampled points. These representations can be regarded as a random sampling of the original dataset. Depending on the employed method, the represented set can be regarded as i.i.d. samples of the original data. However,

methods such as K-means and simple gridding may violate this property due to the uniformity assumption of each abstracted object (e.g., cluster, cell). If the uniformity assumption does not hold in the corresponding region of the original dataset, then the samples are not i.i.d. The significance of the i.i.d. samples with regards to KDE is that it guarantees that the sample set accurately represents the distribution of the original data. In real-world applications, slight representation errors can be tolerated. Therefore, for practical purposes, many of the available summarization techniques can be applied without causing detrimental impact to the resulting estimation quality.

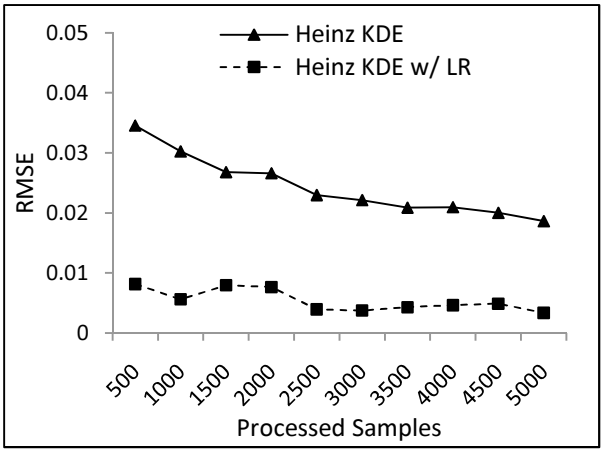
The proposed LR-KDE in Chapter 3 introduces a specific integration of Heinz's clustered sampling approach; however, the above argument implies that the local region concept can be extended to a larger set of stream-based KDEs. The following provides some experimental results to demonstrate the potential impact of LR have on existing stream-based KDEs. The local region bandwidth assignment strategy is applied to four KDE techniques: sample-based KDE with samples = $|D|$ [86], time sample-based KDE [86], Heinz KDE [41, 45, 46], and M-Kernel [98]. For each technique, the local regions and bandwidths are determined at query time. The local region based bandwidths are used for calculating the density estimates. In the preliminary experiment, estimation qualities of the original techniques are compared against their local region extended versions with $|L| = 2$. The test dataset is comprised of a mixture of two Gaussians defined using the canonical form of MIX2. Figure 4.8 shows the results of applying the local region bandwidth scheme to various stream-based KDE techniques.



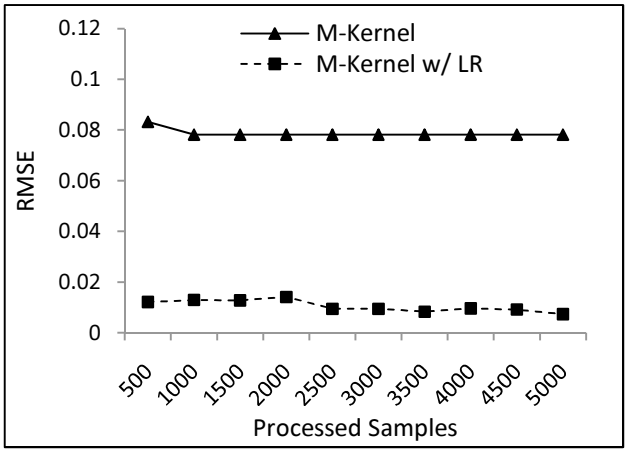
(a) Impact of applying local regions to sample-based KDE



(b) Impact of applying local regions to time-sample-based KDE



(c) Impact of applying local regions to Heinz KDE



(d) Impact of applying local regions to M-Kernel

Figure 4.8 Impact of applying local regions to sample-based KDE

The above experimental results show that the local region bandwidth assignment improves the estimation quality of all the tested stream-based KDEs. Note that in this bimodal dataset, the Heinz KDE and sample-based KDE produced similar estimation quality. This result is an indication of Heinz KDE's excellent clustering performance in retaining the original distributional structure. The results also indicate that the M-Kernel can provide effective summarizations of the data stream. The excellent performance of the local region version M-Kernel would have been difficult to attain if the M-Kernel could not provide an accurate synopsis of the data stream. These results show great promise for the local region based approaches and motivate the development of a generalized algorithm that supports the use of local regions on existing stream-based KDE.

4.7 Conclusion

This chapter provides some theoretical analysis of the LR based KDE. Its expected performances (AMISE) under various conditions are derived. Furthermore, the conditions for which the LR based KDE improves upon the standard KDE are made precise by analyzing the relationship between the aggregated curvature and parameter difference ratio. A discussion on the appropriate local region number is provided that is related to the number of density modes. An alternative criterion for constructing local regions is proposed that can exploit a large set of existing stream-based tools. Lastly, the local region concept is applied to current stream-based KDEs with the initial results showing improved estimation quality in the local region extended approaches.

Chapter 5. Generalized Local Region Algorithm (GLEAM)

Based on the analyses of Chapter 4, we develop the General Local rEgion AlgorithM (GLEAM) to enhance the estimation quality of structurally complex distributions for any existing stream-based KDEs. A set of algorithmic optimizations are designed to enhance the query throughput of GLEAM and to achieve its linear order computation. Additionally, a comprehensive suite of experiments was conducted to validate the effectiveness and efficiency of GLEAM.

Recall that the standard KDE of a fixed sample set, the bandwidth h and kernel function $K(\cdot)$ are adjusted to influence the estimation quality. It has been analytically and empirically shown that the estimation quality of the KDE is critically dependent on the bandwidth while different kernel functions provides marginal effects on the overall estimation quality [39, 80, 82]. In the standard KDE, the bandwidth is applied globally. This global assignment can be problematic for a variety of distributions due to its inability to accurately and precisely model local features. Therefore, for complex densities (e.g., multimodal, long tailed) the global bandwidth KDE may produce poor estimation quality [76, 88]. Due to computational considerations, the majority of stream-based KDEs apply the global bandwidth form. As a consequence, these estimators tend to generate inaccurate estimates of local structures and complex distributions. The emphasis of existing stream-based KDEs is on the development of methods for constructing a finite size model of the data stream. 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, since these techniques apply the standard KDE, they inherit the same estimation issue that is intrinsic to all global bandwidth approaches.

This chapter aims to address the estimation issue associated with the globally assigned bandwidth. In particular, we propose the use of local regions to efficiently and effectively capture local density information in $O(M)$ time. Using local regions, 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 chapter gives the following contributions:

1. **Design of a generalized local region algorithm:** The General Local rEgion AlgorithM (GLEAM) is proposed that can be applied to any existing stream-based KDEs to enhance the estimation accuracy of complex distributions. Analyses of GLEAM's time and space complexity are given.
2. **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.
3. **Comprehensive experiments to validate the effectiveness and efficiency of GLEAM:** GLEAM was applied to a set of representative stream-based KDEs and the results showed that the local region

algorithm improved the estimation quality of existing KDEs. Furthermore, GLEAM produced throughput performances that were comparable or better than its non-local region counterparts.

This chapter is organized as follows: Section 5.1 reviews existing stream-based kernel representation techniques. Section 5.2 describes our proposed GLEAM approach, optimizations, and time and space cost analysis. Section 5.3 gives the experimental results and discussion. Lastly, final conclusions are drawn in Section 5.4.

5.1 Motivation

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

Sample-based and grid-based representation techniques: Sample-based techniques employ a sub-sampling 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 [86]. 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 [91]. 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 [5]. 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.

Cluster-based representation techniques: Cluster-based strategies perform 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 [19, 42]. As a result, the cluster-based estimator does not possess the stability in throughput performance as the sample-based approaches, but can achieve much higher estimation accuracy. Zhou *et al.* introduced the *M-Kernel*, a cluster-based KDE maintenance strategy which performs numerically-based kernel mergers under a fading window [98]. However, their proposed bandwidth strategy does not guarantee that their estimate converges to the true density as the number of samples is increased (i.e., point-wise consistency is not assured). To address the drawbacks of *M-Kernel*, Heinz *et al.* proposed the *Cluster Kernels* which employ a constant time pair-wise merging technique and a global bandwidth scheme based on the Scott's Rule [42]. 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, Boedihardjo *et al.* proposed a cluster-based KDE that supports multiple bandwidth assignments [19].

Need for a Generalized Technique

All of the above stream-based KDE approaches are dependent on a specific 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 brethrens. 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 towards 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 upon other characteristics (e.g., sample throughput). Hence, this paper 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.

5.2 GLEAM Approach

The analyses of Chapter 4 demonstrated 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. The significance of the i.i.d. samples is that it guarantees that the sample set accurately represents the distribution 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. However, methods such as gridding may violate this property due to the uniformity assumption associated with each summarized object. If the uniformity assumption does not hold in the corresponding region of the true distribution, then the gridded objects are not identically distributed. However, such representation errors can often be parameterized and bounded to an arbitrarily small error ε. This fact implies that the local region bandwidth scheme can be applied to any existing stream-based KDEs given that the representative sampling components accurately model the distribution of the original dataset.

The discussion of Sections 4.4.2 and 4.4.3 demonstrates that the local region based KDE can provide improved estimation accuracy over the standard KDE. Furthermore, Section 4.6 shows that the local region scheme can be applied to existing stream-based KDEs to enhance their estimation quality. This characteristic of the local region scheme motivates the development of the Generalized Local rEgion AlgorithM (GLEAM). GLEAM is a local region 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}.

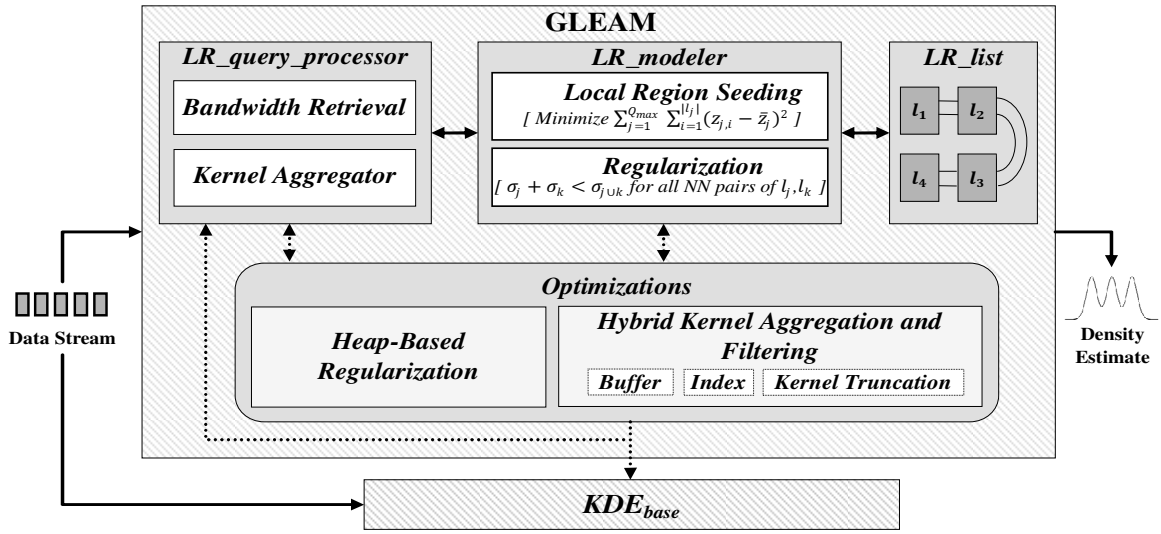


Figure 5.1 GLEAM Architecture

Figure 5.1 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 local regions for the current stream; *LR_modeler* directs the construction of the local regions 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: local region construction and density query processing. For local region construction, the *LR_modeler* continuously maintains a set of local regions stored in *LR_list* which 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 their original approach in maintaining the M representative kernel objects. For the *LR_modeler*, the sample is processed to continuously maintain a set of local regions which describes the current stream. This process of GLEAM is known as local region construction which is discussed in Section 5.2.1. To generate a density query estimate, the *LR_query_processor* module retrieves the relevant local regions 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 5.2.2. Cost analyses and optimizations techniques of GLEAM are provided in Sections 5.2.3 and 5.2.4, respectively.

5.2.1 Local Region Construction with *LR_Modeler*

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

Local region construction objective criterion

To construct the local regions, a criterion is proposed based on the application of the Scott's Rule to each local region. Imposing this bandwidth results in an error form (see Section 4.4.2) 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 the AMISE (), the following local region assignment heuristic is proposed: each local region 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 local region is designed to model a unimodal density, then the construction of the local regions should aim to minimize the total sum squared error (SSE). If the SSE minimization objective is employed, then the number of local regions Q must be determined. Since each local region is aimed to capture a unimodal structure, the natural choice for Q is the number of true modes in the density. Verifying against Equation 4.42, this choice of Q will give a relatively low numerator value and high denominator value in $\beta(K, L, D)$.

To estimate the number of modes within the density, σ_j is used to measure the suitability of the unimodality assumption imposed on local region l_j. Suppose that a representative sample set D of the density is partitioned into r number of local regions constructed via the SSE criterion. Let σ_j and σ_k be the deviations of the nearest neighbor pair of local regions l_j and l_k, respectively. Furthermore, define $\sigma_{j \cup k}$ to be the deviation of the merged pair $l_j \cup l_k$. The nearest neighbor of l_j is defined to be the local region that has a center that is 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 local regions. 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 local regions $s \leq r$ would be a suitable estimate for the number of modes in the sample set D.

Based on the above mode estimation strategy, the *LR_modeler* generates the local region in two phases: local region seeding and local region regularization. In local region seeding, the *LR_modeler* continuously constructs and maintains Q_{max} number of local regions in the *LR_list* which represents the data stream with a high level of granularity. In the second phase, the *LR_modeler* regularizes the local regions in *LR_list* by merging those regions that invalidate the unimodality assumption as described above. The result of this regularization phase will guarantee that any nearest neighbor pair of local regions (l_j, l_k) satisfies $\sigma_{j \cup k} > \sigma_j + \sigma_k$. The combined local regions are used to process the density estimates.

Phase 1 Local region seeding

The process of local region seeding employs a single-scan incremental K-Means clustering on the data stream [7]. Let Q_{max} be the maximum number of local regions that can be stored in *LR_list* to provide a fine granular representation of the data stream. Each local region 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 $\text{center}(l_j)$ and $\text{radius}(l_j)$ can be quickly computed from the components of $\vec{v}_j$. If the local region vector addition/subtraction is defined as the component-wise addition/subtraction operation, then the local regions can be efficiently merged and deleted using these vector operations.

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 $\left(d \cap \left[\left(\text{center}(l_j) - \mu \cdot \text{radius}(l_j)\right)..\left(\text{center}(l_j) + \mu \cdot \text{radius}(l_j)\right)\right]\right) \neq \emptyset$ and where $\mu \geq 1$. However, if the *LR_modeler* cannot find an intersecting l_j, a new local region is created for d. The construction process continues in this manner until the number of local regions is greater than Q_{max} (i.e., $|L| > Q_{max}$). When $|L| > Q_{max}$, the *LR_modeler* merges the two nearest neighbor local regions (as defined above) that minimize the SSE residual. The merging of any two local regions 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 Local region regularization

The number of local regions Q_{max} in the *LR_list* can be much larger than the actual number of modes in the density. This condition can cause certain local regions 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 local regions l_j and l_k that have $\sigma_{j \cup k} \leq \sigma_j + \sigma_k$. The nearest neighbor local region pair is defined as the two local regions with the minimum L_2 distance between their centroids from all other local region pair sets.

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

5.2.2 Query Processing with *LR_query_processor*

To process a density query, the *LR_query_processor* initiates the local region 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 local regions 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 $BQ(x) = H_{LR}(l) \mid l \in L_{reg} \wedge (x \cap (\Omega(l)..\Theta(l)] \neq \emptyset)$ where $x \in \mathcal{R}$ and $H_{LR}(l)$ is the bandwidth of local region 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 4.2.

To further illustrate the density estimation algorithm, an example is provided as follows. 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 local region 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 $BQ(\text{K\_center}(k))$ 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 is aggregated following the form as dictated by $\hat{f}_{LR}(\cdot)$ in Equation 4.2. 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 $BQ(\cdot)$, since 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 optimization (Section 5.2.4) which can reduce the density query cost to $O(|M - T| \cdot \log(Q_{max}))$ where $T \geq 0$.

5.2.3 Time and Space Cost Analyses

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

Local region construction

The local region construction involves two subtasks: local region seeding and local region regularization. These subtasks impact both the insertion and density query costs. Since local regularization is invoked only at query time, the insertion cost (without consideration to KDE_{base}) is solely dependent on the local region seeding. The cost of the local region seeding is equal to the combined costs of locating/inserting an intersecting local region ($O(Q_{max})$) and merging a pair of local regions ($O(Q_{max})$). 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 $O(Q_{max}) + \text{insert\_cost}(KDE_{base})$. The insertion cost of the KDE_{base} is bounded by $O(M)$. Because $Q_{max} \ll M$ and Q_{max} is 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 local region regularization phase, a copy process from *LR_list* to L_{reg} is $O(Q_{max})$ and each scan/merge is bounded by $O(Q_{max})$. The algorithm performs at most $O(Q_{max})$ 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 $O(M)$.

5.2.4 Optimizations

In the following, two sets of optimization strategies are proposed to enhance the performance of the local

region regularization operation and to reduce 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 local region merge candidates. This first set of optimization 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 which includes techniques that perform local region buffering, local region indexing, and kernel truncation.

Heap-based Regularization Optimization

To reduce the computation of the regularization task, an optimization scheme is proposed that performs incremental breadth first search strategy to merge the closest candidate local region pairs. Initially, the elements of *LR_list* is copied to the local region list structure L_{reg}. Next, each pair of local regions in L_{reg} which 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 local regions is dequeued from the heap. Because the dequeued object pair represents the closest local regions that satisfy $\sigma_{j\cup k} \leq \sigma_j + \sigma_k$, the pair is merged into a single local region 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 local region l_m and its current nearest neighbors, then these candidates along with l_m are re-inserted 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 local regions 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 local regions 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 the manner described above until there are no more candidates available (i.e., heap is empty). At the algorithm's termination, L_{reg} will contain the regularized local regions where $|L_{reg}| \leq |L|$ and each nearest neighbors 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 $O\left(Q_{max} + {Q_{max}}^2\right)$ cost of the direct method described in Section 5.2.1. Integrating this optimization strategy results in lowering the density query cost to $O(Q_{max} + Q_{max}\log(Q_{max}) + \sum_{i=1}^{M} \mathrm{Cost}\left(BQ(m_i)\right)$ 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 local region search and kernel aggregation. These techniques are

integrated to provide the *hybrid kernel aggregation and filtering* optimization.

The first strategy, *local region buffering*, stores previously accessed local region 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) local region 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 5.2.2. 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 local regions and therefore would benefit from the cached local region. 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, *local region indexing*, replaces the L_{reg} linked list with a pre-allocated array structure. Because the output of the local region regularization process maintains the sorted order of the regions' centroids, the local regions can be transformed to a pre-allocated array structure that preserves this ordering in linear time. With the pre-allocated array, binary searches can be performed to locate a local region 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 additional errors can be introduced by the truncated kernels, a bound on these errors derived parameterized via a user-defined parameter p:

Equation 5.1

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

where m_i is a kernel object in KDE_{base}, $w(\cdot)$ is the weight, h_{min} is the minimum bandwidth in L_{reg}, p 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 $|x - m_i| \geq ph_{min}$.

With the truncated kernels, the *LR_query_processor* determines in $O(Q_{max})$ time the boundaries of kernel objects which will have contributions below the specified threshold (i.e., determine kernel centers with distance $|x - m_i| \geq ph_{min}$). 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 local region buffering, local region 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 $O(Q_{max} + Q_{max}\log(Q_{max}) + |M-T| \cdot \log(Q_{max}))$.

5.3 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, and estimation under concept drift. The experiments are organized as follows: Section 5.3.1 introduces the experiment design. Sections 5.3.2 - 5.3.4 provide an in-depth evaluation of GLEAM's effectiveness when applied to existing stream-based KDEs under the following metrics: estimation accuracy, sample throughput, and query throughput. In Section 5.3.5, GLEAM's optimization strategies are analyzed using various baseline KDEs and datasets. Lastly, Section 5.3.6 summarizes the experimental results.

5.3.1 Experiment Design

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

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 which were constructed from mixture of normals: MIX2, MIX4, and MIX8. For each mixture, 5 time series instances were created. Hence, there are a total of 15 synthetic time series datasets. The real-world data consist of highway loop detector measurements (HIGHWAY) and power demand log from a Dutch facility (POWER). 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. For further details on the dataset, please see Appendix A.

Algorithms and Parameters

Five existing global bandwidth stream-based density estimators were evaluated including: (1) Epanechnikov KDE (EKDE) [86]; (2) Gaussian KDE (GKDE); (3) Cluster Kernels KDE (CKKDE) [42]; (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 5.1, the Cluster Kernels KDE (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 non-optimized 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 [80]. For the GLEAM based KDEs, the following parameters were used: $Q_{max} = 10$, $M = 1000$, and each local region applied the Scott's Rule bandwidth.

Evaluation Criterions

The integrated absolute error (IAE) is employed to quantify the estimation discrepancy. Here, IAE is defined as follows:

Equation 5.2

$$IAE(\hat{f}) = \sum_{i=0}^{1000} ABS\left(\hat{f}(x_i) - f(x_i)\right)\Delta$$

where x_1.. x_{1000} are query points which uniformly divide the entire span of the distribution and $\Delta = \frac{x_T - x_1}{1000}$.

Hence, accuracy is defined as the normalized complement of IAE (i.e., difference between the theoretical maximum IAE and the computed IAE normalized by the theoretical maximum IAE). The following provides the formula of the accuracy score:

Equation 5.3

$$NCI(\hat{f}) = 1 - \left(\sum_{i=0}^{1000} ABS\left(\hat{f}(x_i) - f(x_i)\right)\Delta\right)/E_{max}$$

where x_1.. x_{1000} are query points which uniformly divide the entire span of the distribution, $\Delta = x_{i-1} - x_i$, 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 the above 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 2.0 GHz Intel Pentium Core 2 Duo with 3 GB of RAM.

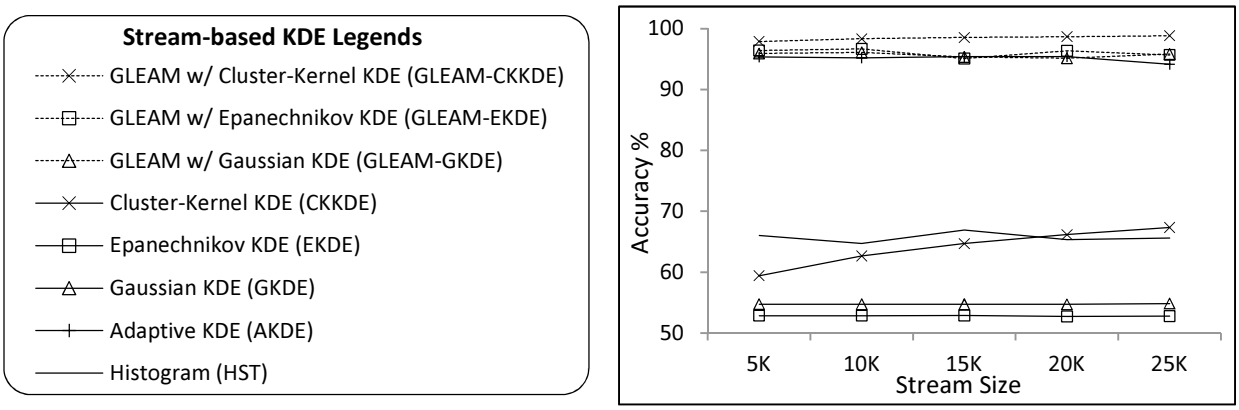
5.3.2 Estimation Accuracy

The results of the estimation accuracy are provided in Figure 5.2. In the graphs, the *x-axis* is the sample size and the *y-axis* is the estimation accuracy (Equation 5.3). 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

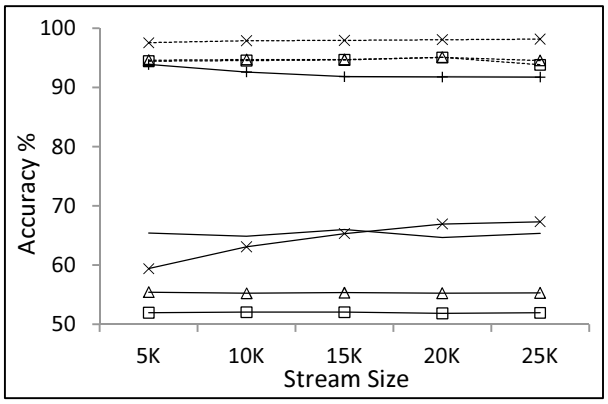
KDEs were with the MIX2, MIX4, and MIX8 datasets which provide substantial improvements (23% to 44%) over the global bandwidth KDEs (i.e., CKKDE, EKDE, and GKDE). These large gains can be attributed to the datasets' high level of structural complexity (MIX4 and MIX8) and highly localized features (MIX2). 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 exceed 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 GLEAM based KDEs consistently improved the accuracy of GKDE and EKDE (for data size ≥ 10K). For data sizes 10K to 20K, the GLEAM-CKKDE provided comparable scores to the CKKDE and slightly lower (1.4%) for 5K and 25K data sizes.

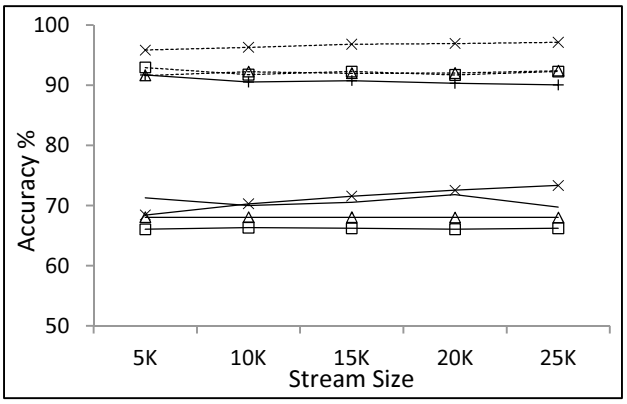
In summary, the GLEAM algorithm improved the accuracy in most of the datasets and exhibited the highest gains for complex densities (MIX2, MIX4, MIX8, and POWER). The GLEAM based KDEs also inherit the advantages of its base KDE which can further improve its estimation accuracy. The standard deviation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



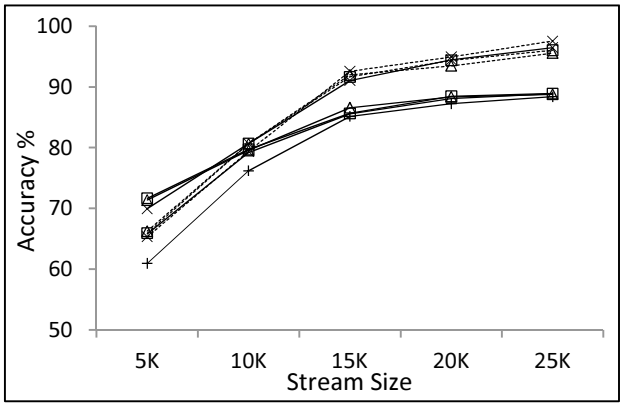
(a) Estimation Accuracy of MIX2 Data



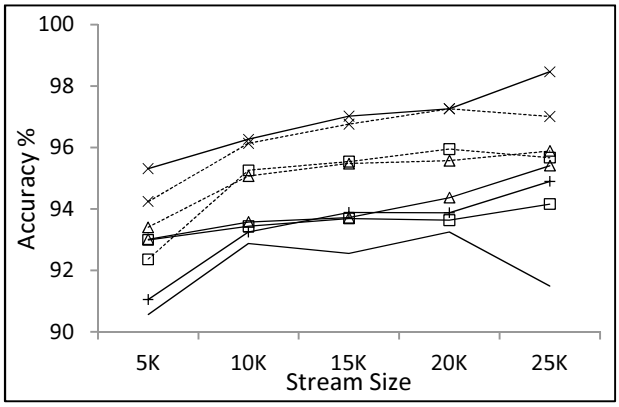
(b) Estimation Accuracy of MIX4 Data



(c) Estimation Accuracy of MIX8 Data



(d) Estimation Accuracy of POWER Data

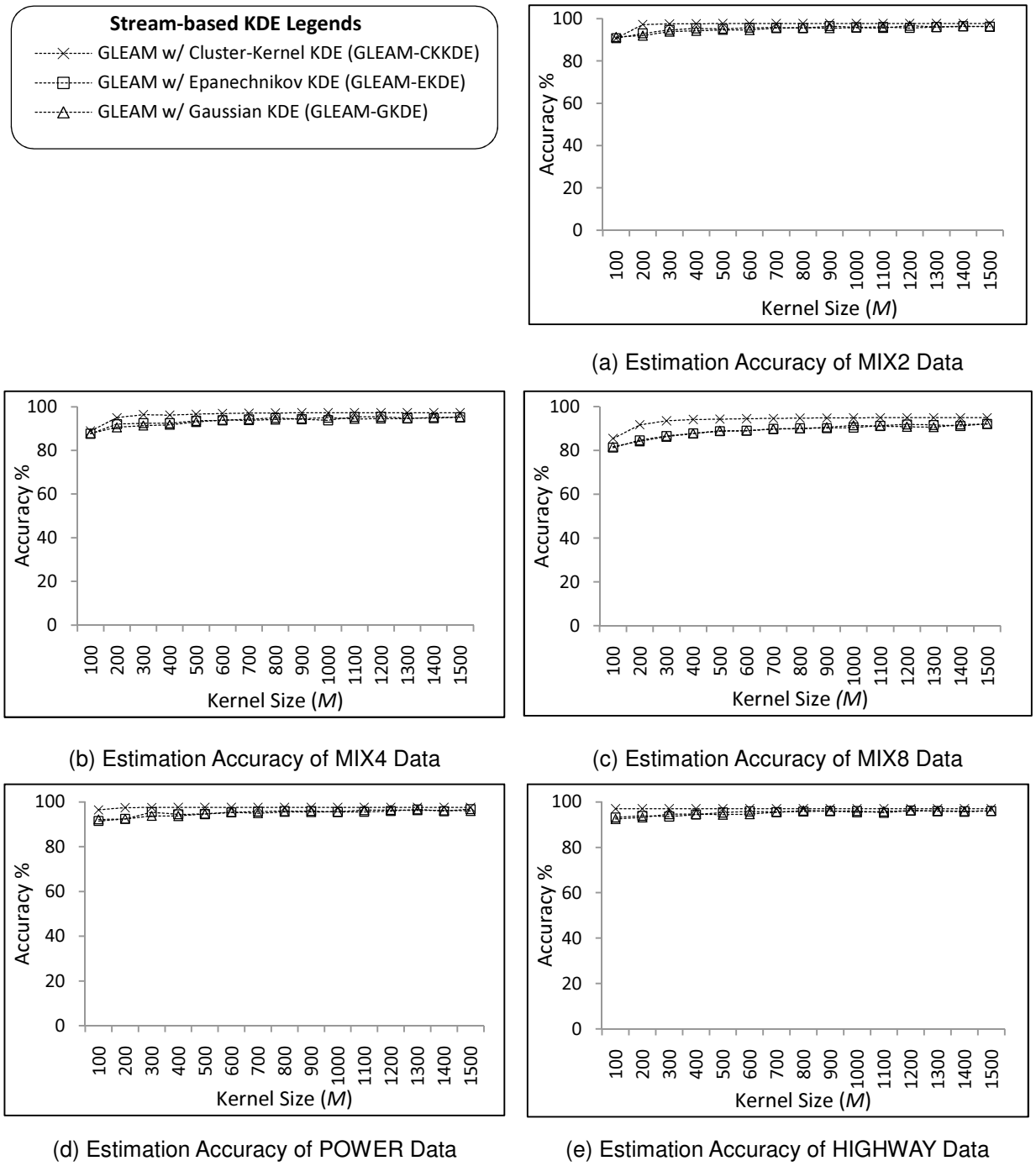


(e) Estimation Accuracy of HIGHWAY Data

Figure 5.2 Estimation accuracy of all the Datasets (note the varied accuracy scale)

5.3.2.1 Sensitivity Analysis: Effects of Kernel Size and Bin Size

This section investigates the effects of GLEAM's bin size and kernel size on estimation quality. Figure 5.3 shows the results of the estimation accuracy for a range of kernel sizes from 100 to 1500 for all the datasets. There are two key observations from these results: (1) the accuracy improves as more kernel samples are employed and (2) and there are diminishing returns as more samples are considered. The first observation is consistent with the asymptotic consistency analyses of Sections 4.2 and 4.3 where improved estimation accuracy can be expected as more samples are integrated into the kernel density model. The second observation supports the derived convergence rate of Section 4.4.1 where the reduction in error diminishes as the sample size is increased. Overall, the estimators are able to attain stable error rates throughout the given range of kernel sizes. These results also help validate the cluster-based approach's (i.e., GLEAM-CKKDE) ability to support more accurate data representation than the sample-based approaches as the cluster-based estimators consistently attain higher accuracy scores the sample-based techniques.



(a) Estimation Accuracy of MIX2 Data

(b) Estimation Accuracy of MIX4 Data

(c) Estimation Accuracy of MIX8 Data

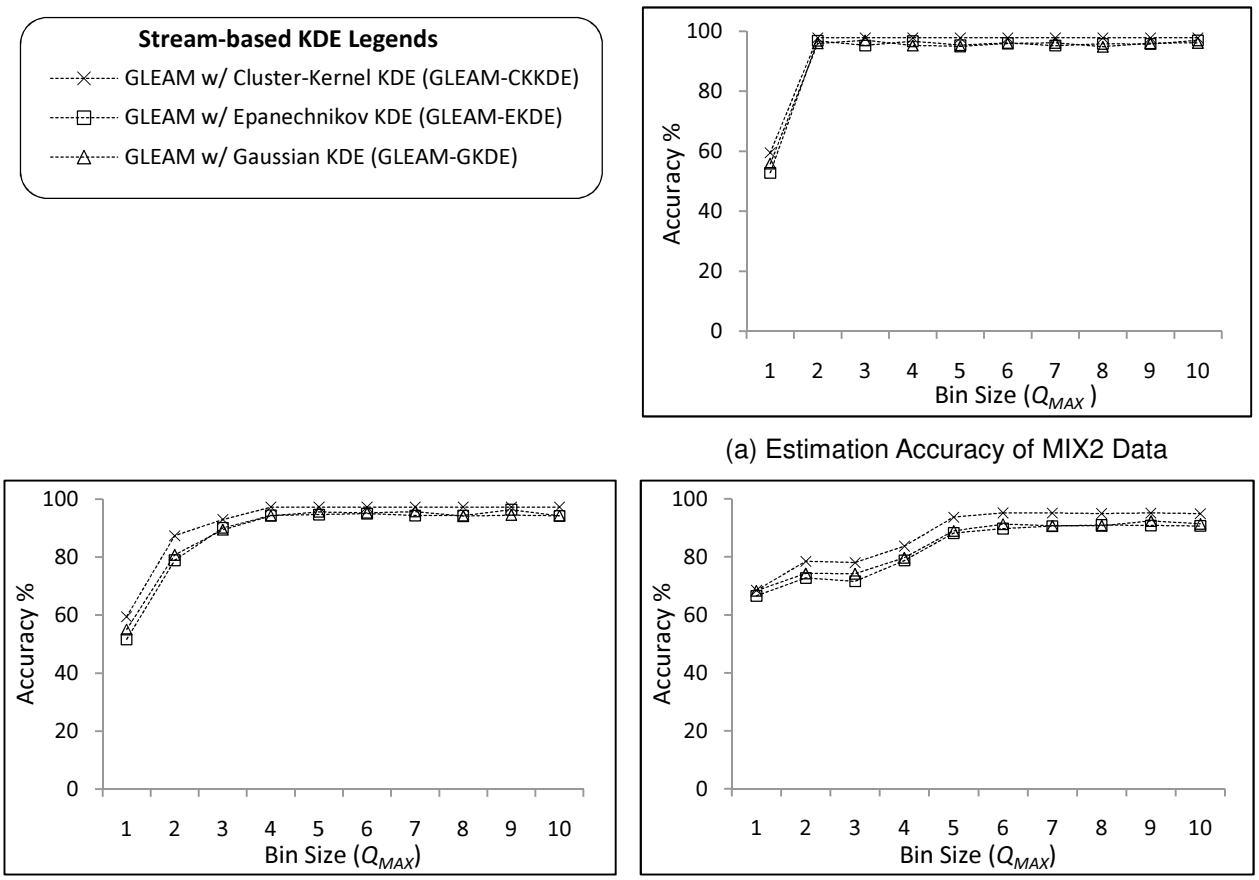
(d) Estimation Accuracy of POWER Data

(e) Estimation Accuracy of HIGHWAY Data

Figure 5.3 Estimation accuracy based on varying kernel size

Figure 5.4 shows the results of estimation accuracy based on varied bin sizes from 1 to 10. Overall, the estimation accuracy improves and stabilizes as the number of bin size is increased. For all the datasets, the estimation accuracy sharply increases in the lower bin size range (from 1 to 5) due to the insufficient number of bins required to model the datasets. As the bin size is increased, the accuracy scores tend to

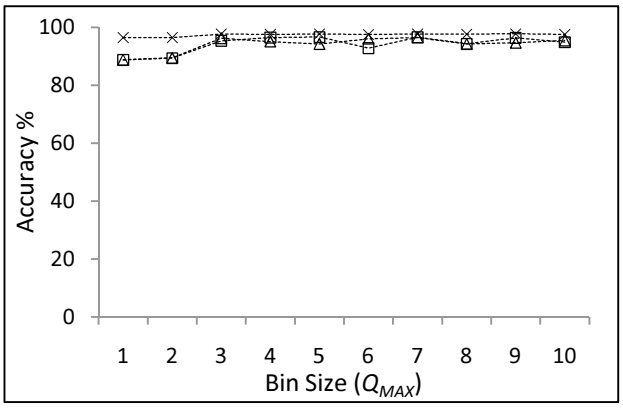
flatten and stabilize since the larger bin size can capture the significant features (e.g., modes) of the distributions. Although the bin size can be set to an arbitrarily large number, the GLEAM approach will iteratively merge the bins and reduce its size (via regularization described in Section 5.2.1 Phase 2) until the minimum number of bins (i.e., effective bin size) is achieved which can sufficiently model all of the distributional features. For example, the effective bin size after regularization in MIX2 is 2 for all bin sizes greater than 1. The results of this experiment justify the use of a sufficiently large bin size to model a given dataset provided that the effective bin size $\leq Q_{max}$.



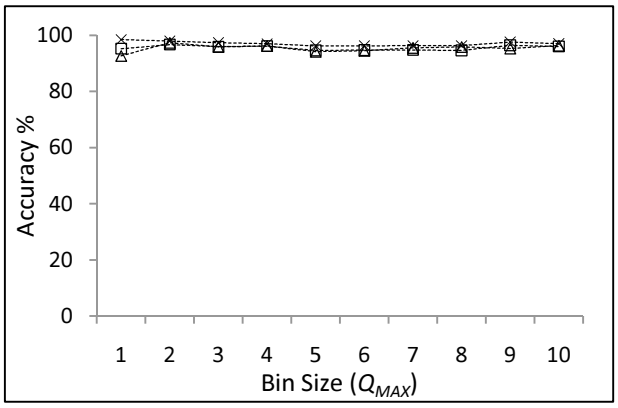
(a) Estimation Accuracy of MIX2 Data

(b) Estimation Accuracy of MIX4 Data

(c) Estimation Accuracy of MIX8 Data



(d) Estimation Accuracy of POWER Data

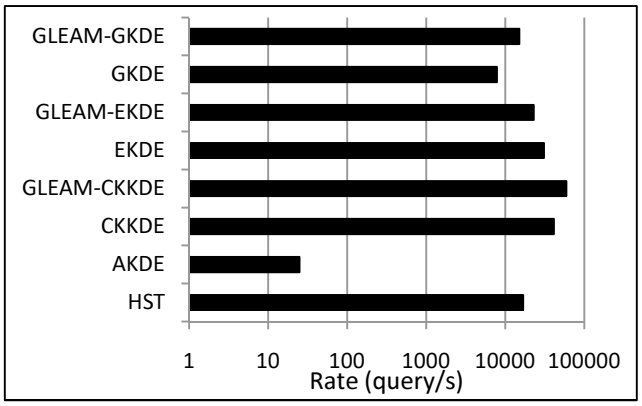


(e) Estimation Accuracy of HIGHWAY Data

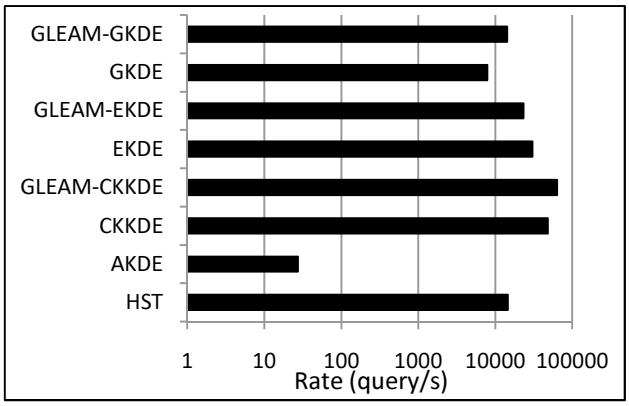
Figure 5.4 Estimation accuracy based on varying bin size

5.3.3 Query Throughput

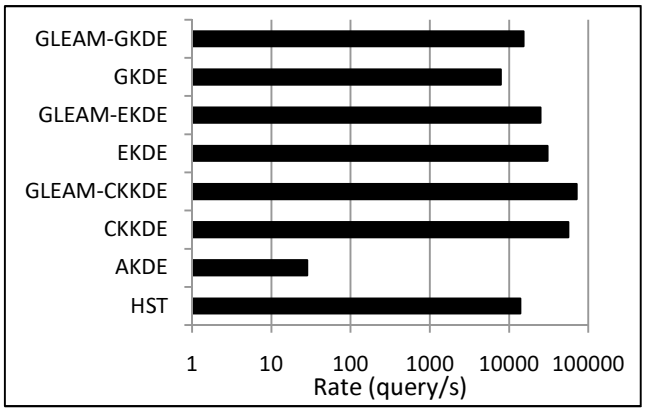
Figure 5.5 gives the query throughput results of all the estimators. The *x-axis* is the query throughput (query/sec) and the *y-axis* is the density estimator. In most instances, the optimal GLEAM based KDEs provided higher query throughput over their base KDEs due to GLEAM's ability to efficiently regularize the local regions 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 which 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 10^{-3} rate of the next slowest estimator). The standard deviation percentage for this set of experiment is $\leq 7\%$.



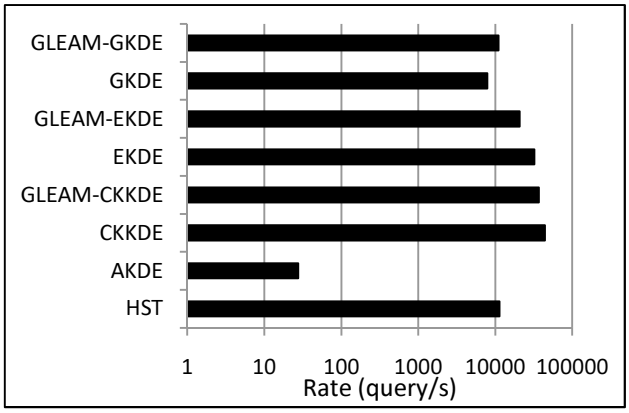
(a) Query Throughput of MIX2 Data



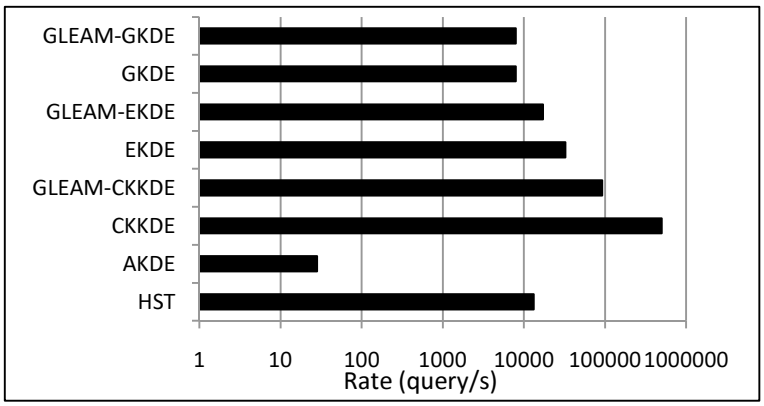
(b) Query Throughput of MIX4 Data



(c) Query Throughput of MIX8 Data



(d) Query Throughput of POWER Data

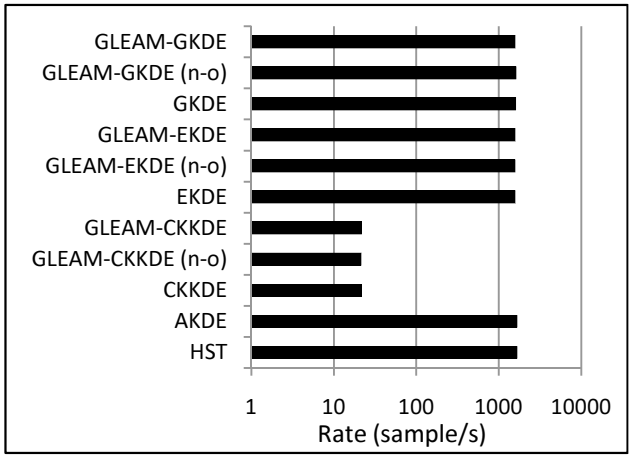


(e) Query Throughput of HIGHWAY Data

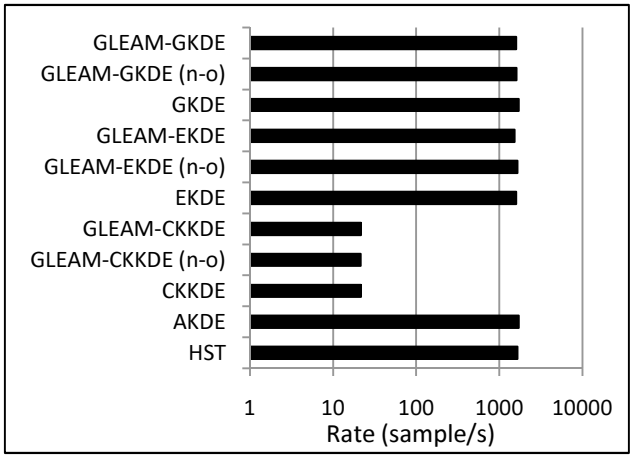
Figure 5.5 Query Processing Performance (log scaled)

5.3.4 Sample Throughput

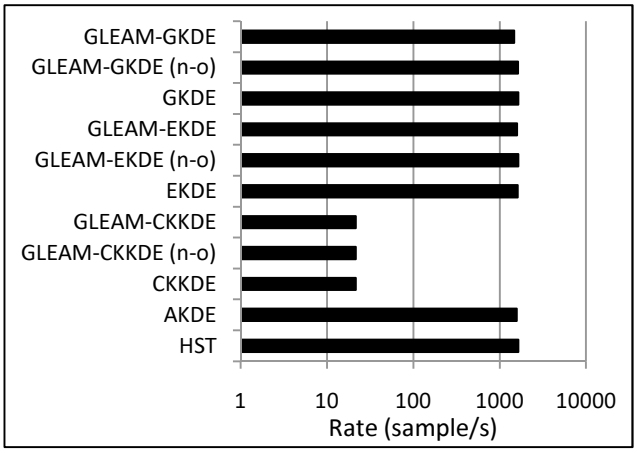
The sample throughput is given in Figure 5.6. In the plots, the *x-axis* is the sample throughput rate (sample/sec) and the *y-axis* is the KDE technique. Here, "**n-o**" refers to the non-optimized GLEAM based KDEs. The GLEAM based KDEs produced negligible overhead in the sample throughput. In fact, most of the differences between the (optimized and non-optimized) 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 where it achieved exceptionally high throughput. This is due to the significantly lower number of discrete values in the HIGHWAY dataset (≤ 100) which 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 shows that GLEAM can successfully adopt the differing capabilities of the base KDEs. The standard deviation percentage for this experimental component is $\leq 7\%$.



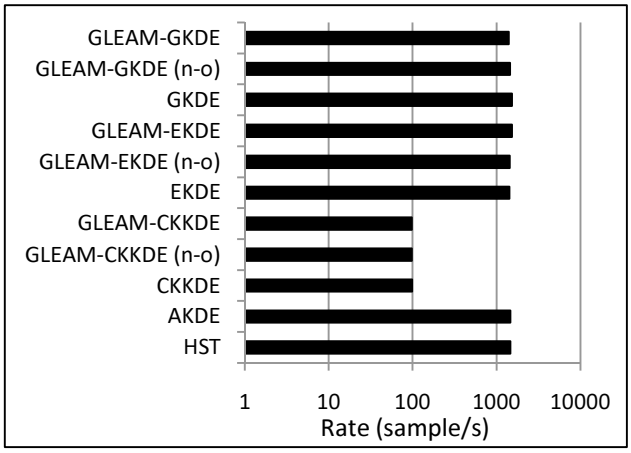
(a) Sample Throughput of MIX2 Data



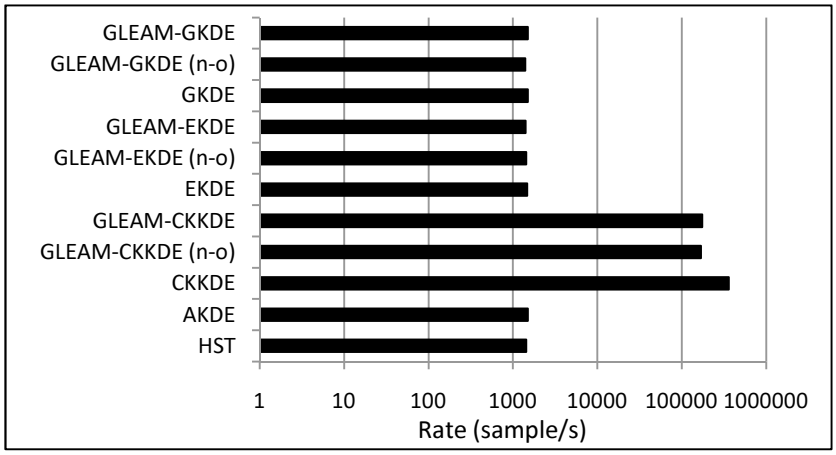
(b) Sample Throughput of MIX4 Data



(c) Sample Throughput of MIX8 Data



(d) Sample Throughput of POWER Data

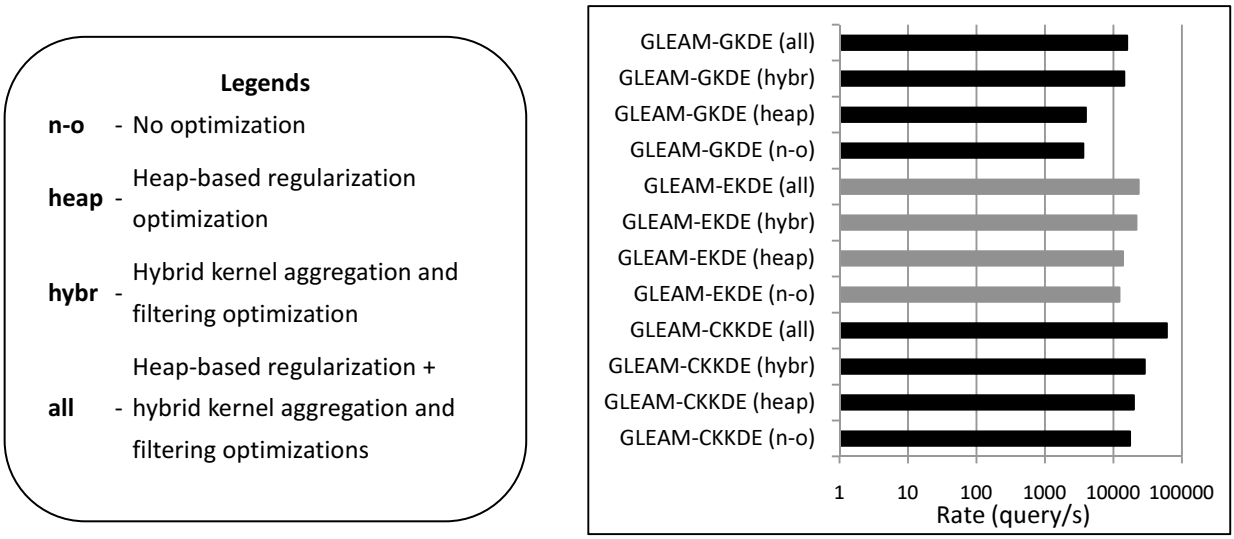


(e) Sample Throughput of HIGHWAY Data

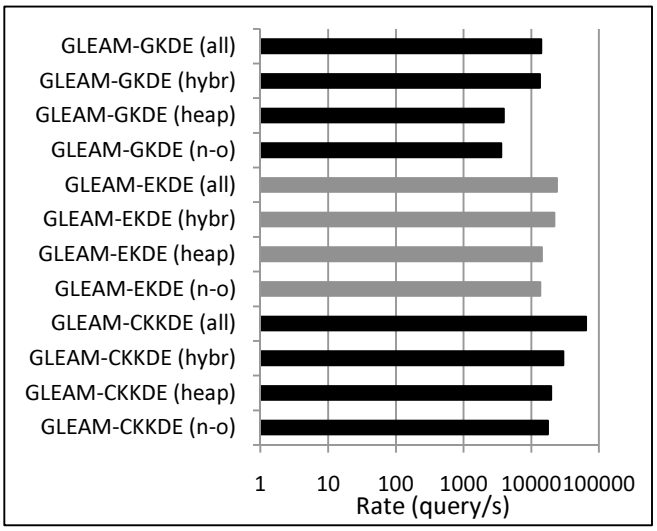
Figure 5.6 Sample Processing Performance (log scaled)

5.3.5 Optimization Effectiveness

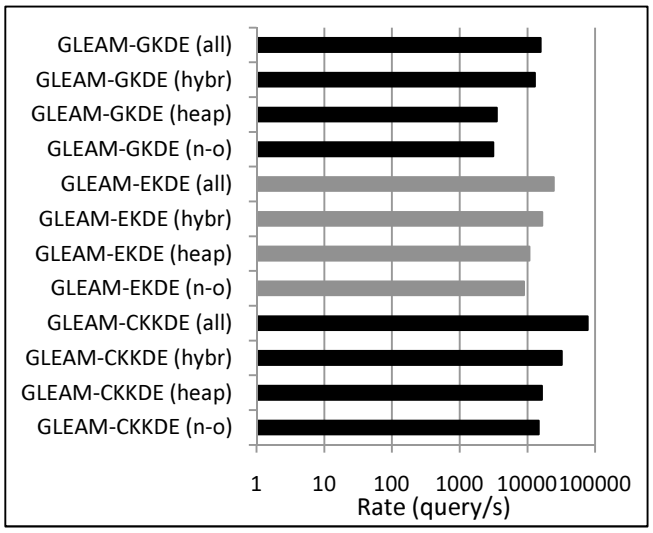
This experiment tests the effectiveness of the proposed optimization strategies: *heap-based regularization* and *hybrid kernel aggregation and filtering*. Figure 5.7 shows the query throughput of GLEAM with varying combination of optimization techniques. The *x-axis* is the query throughput (query/sec) 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 MIX2, MIX4, and MIX8 datasets, the *hybrid kernel aggregation and filtering* optimization provides the highest increase in query throughput for GKDE than 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.



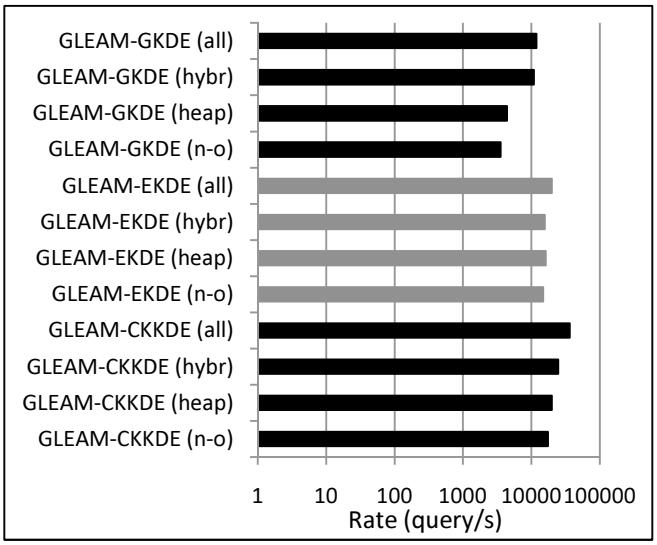
(a) Query Throughput of MIX2 Data



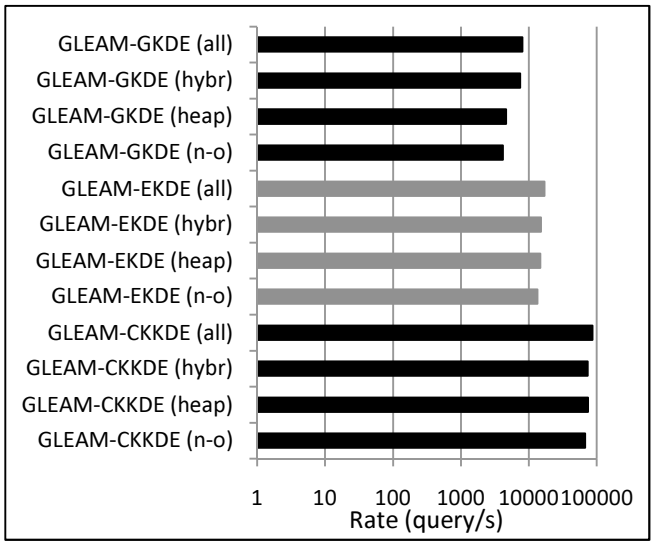
(b) Query Throughput of MIX4 Data



(c) Query Throughput of MIX8 Data



(d) Query Throughput of POWER Data



(e) Query Throughput of TRAFFIC Data

Figure 5.7 Query Processing Performance of Various Optimizations (log scaled)

5.3.6 Discussion

The experiment in Section 5.3.2 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 local region regularization, GLEAM has 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 fail detection 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 5.3.3 - 5.3.5, 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 in the HIGHWAY dataset. In this dataset, GLEAM-CKKDE outperformed all the non clustered-based KDEs in query throughput while 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.

5.4 Conclusion

In this chapter, the generalized local region based algorithm (GLEAM) framework is proposed that can be applied to existing stream-based KDEs to enhance the estimation accuracy of structurally complex distributions. Consistent with the theoretical analyses of Chapter 4, 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 resulted 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.

Chapter 6. Applications: Online Outlier Detection

This chapter describes two applications of the density estimator for the problem of online outlier detection. The first application is a novel probabilistic and multi-scale outlier detection framework to capture both point and clustered outliers. The second application is the development of an outlier detection scheme for automatic discovery of roadway traffic incidents. This chapter is organized into two sections: Section 6.1 describes the work of the probabilistic and multi-scale outlier detection framework and Section 6.2 discusses the proposed approach for automatic detection of traffic incidents.

6.1 Application I: Multi-scale Outlier Detection Framework

In this work, an outlier detection framework is proposed which can discover outliers in multiple time scales. The multi-time scale approach effectively detects of both point and clustered outliers under a set of multiple sliding windows. The unique features of the proposed approach are: (1) the construction of a generalized multi-scale representation of distributional patterns via hierarchical aggregation; (2) and the unification of both point and clustered outlier detection; (3) the integration of non-parametric LR-KDE to model the background density structure. This section is organized as follows: Section 6.1.1 provides the motivation. Section 6.1.2 summarizes the contribution. Section 6.1.3 details the proposed outlier detection framework. Section 6.1.4 presents the experimental results. Lastly, Section 6.1.5 gives the conclusion.

6.1.1 Motivation

In time indexed data streams, outliers are functions of the time and scale of the observation [17, 94]. For example, in highway traffic data, a particular event such as congestion can be classified as an outlier (e.g., non-recurrent congestion) with respect to a background traffic pattern of the last one hour. However, if the traffic observation is taken to be the last 24 hours, the same event can be classified as normal (e.g., recurrent congestion). Additionally, an outlier can exist as an isolated point or as a group/cluster of outlying points. Using the traffic example above, if a significant incident occurred within the last one hour window, the entire set of observations of that window is potentially an outlier within the background pattern of the last 24 hours. Hence, to effectively detect both point and clustered outliers within varying sets of time spans, a multi-scale outlier detection approach is proposed. In this work, the problem of detecting both point and clustered outliers residing in different time windows (e.g., one hour, 24 hour) is transformed into a problem of detecting point outliers in multiple time scales. This work proposes a multi-scale outlier detection framework which employs multiple sliding windows (of varying time length and scale) to detect both point and clustered outliers. Using this approach, clustered outliers from the finest/lowest time scale (e.g., raw data stream samples) are represented as points in a higher time scale which can be uncovered using point-based outlier detection techniques.

6.1.2 Contribution

The contribution of this work includes the following:

1. **Multi-scale outlier detection framework –** Design of an outlier detection framework which discovers outliers at multiple scales, detects both point and clustered outliers via a unified testing methodology, and supports the use of an arbitrary density estimation technique.
2. **Enhanced LR-KDE –** Extended the LR-KDE approach by integrating GLEAM's local region construction and regularization algorithms to enhance the robustness of density estimates.
3. **Empirical validation –** Conducted a comprehensive set of simulations to test and compare the effectiveness of the outlier detection framework for a variety of density estimators.

6.1.3 Proposed Approach: Multi-scale Outlier Detection

This subsection discusses the approach of the multi-scale outlier detection method. A formal problem statement is provided in the following: *Let* $D = \{x^{(1)}, x^{(i)}, .., x^{(n)}\}$ *be a data stream where* $x^{(i)}$ *is a real-valued scalar and each* $x^{(i)}$ *is associated with a timestamp* $\tau(x^{(i)}) \in [\tau_{min}, \tau_{max}]$. *Furthermore, let* $W^{(l)}_{\tau_\alpha,\tau_\beta}$ *be a time-based sliding window on* D *where* $0 < l \leq \mathrm{Z}$ *is the window scale level,* $\tau_\alpha \in [\tau_{min}, \tau_{max}]$ *is the start time, and* $\tau_\beta \in [\tau_{min}, \tau_{max}]$ *is the end time. Define the construction of* $W^{(l)}_{\tau_\alpha,\tau_\beta}$ *as follows:* $W^{(l)}_{\tau_\alpha,\tau_\beta} = \{G\left(W^{(l-1)}_{\tau_\alpha,\tau_{\alpha+1}}\right), \ldots G\left(W^{(l-1)}_{\tau_i,\tau_{i+1}}\right), \ldots G\left(W^{(l-1)}_{\tau_{\beta-1},\tau_\beta}\right)\}$ *where* $G(W)$ *is an aggregation operator on W,* $\tau_i < \tau_{i+1}$, *and* $W^{(1)}_{\tau_\alpha,\tau_\beta} \subseteq D$. *Given a threshold* Γ, *a window sample* $o \in W^{(l)}_{\tau_\alpha,\tau_\beta}$ *is an outlier if the probability w.r.t to window* $W^{(l)}_{\tau_\alpha,\tau_\beta}$ *is* $P_{W^{(l)}_{\tau_\alpha,\tau_\beta}}(o - \varepsilon \leq o \leq o + \varepsilon) \leq \Gamma$ *where* ε *is the interval radius. The task is to determine all such outliers within windows* $W^{(1)} \ldots W^{(\mathrm{Z})}$.

As described by the above problem statement, outliers are dependent upon the set of windows $W^{(1)} .. W^{(\mathrm{Z})}$ that represent multiple time scales. By defining outliers with respect to these multi-scale and multi-length windows, both point and clustered outliers are determined by employing a *point-based* probability estimates within each window. For example, at the lowest scale window $W^{(1)}$, probability estimates of incoming data samples $x^{(i)}$ can be generated and compared against threshold Γ to determine point outliers. At a higher level window $W^{(l>1)}$, the same point probability estimation strategy is invoked to determine outlying sample objects. Because the sample points $o \in W^{(l>1)}$ are aggregates of $W^{(l-1)}$, an outlier found in $W^{(l>1)}$ is an outlying object composed of (potentially) multiple elements from D (i.e., clustered outliers).

In the proceeding section, the operations involved within each hierarchy are discussed in detail. An

overview of the multi-scale outlier detection framework is given in Section 6.1.3.1. A discussion of the *density modeling* is provided in Section 6.1.3.2. Next, the *window aggregation* operation is described in Section 6.1.3.3. Lastly, the *outlier testing* component is discussed in Section 6.1.3.4.

6.1.3.1 Overview

The multi-scale online outlier detection framework is provided in Figure 6.1below. Within the multi-scale outlier detection framework, the finest scale (i.e., raw stream data) resides at the bottom-most hierarchy (level = 1) and the highest (most coarse) scale are represented in the top-most hierarchy (level = Z). The double solid lines indicate windows and operations that are currently active and dashed lines denote windows and functions that have been processed. Due to the use of an arbitrary aggregation operator $G(W)$, each window in $W^{(1)}$... $W^{(\mathrm{Z})}$ are explicitly modeled and maintained. Hence, at given time the proposed outlier detection framework will retain Z number of sliding windows where the *window aggregation* and *outlier testing* operations are performed at the end of each sliding window. In general, the window sample elements of a given hierarchy are derived from the aggregation of windows from the lower hierarchy. Within each hierarchy, three major tasks are performed:

1. **Outlier testing –** performs a test on the newly incoming sample object to determine its outlying condition.
2. **Density modeling –** constructs a probabilistic density model of the current active window.
3. **Window aggregation –** summarizes the window elements into a set of sample objects to be integrated into the active window of the upper hierarchy.

Furthermore, because the aggregated windows within a given hierarchy do not overlap, aggregated results are disjoint on time. Hence, window aggregation and outlier testing are invoked only when the sliding window has purged all of the samples from the previous aggregation.

An example execution trace of the multi-scale outlier detection is provided as follows: From the bottom-most level, an incoming data stream element x is tested against the current active window $W^{(1)}_{\tau_{\beta-1},\tau_\beta}$ to determine its outlying condition. If it is an outlier then a copy of the sample is forwarded to the *outlier set*. After completing *outlier testing* and regardless of the result, x is forwarded to the current active window $W^{(1)}_{\tau_{\beta-1},\tau_\beta}$ to update its density model. The outlier tests and window updates are continuously invoked until a complete window length slide is performed i.e., $W^{(1)}_{\tau_{\beta-1},\tau_\beta}$ becomes $W^{(1)}_{\tau_\beta,\tau_{\beta+1}}$. At this point, the previous window $W^{(1)}_{\tau_{\beta-1},\tau_\beta}$ is *aggregated* and the aggregated results tested for outliers against the active window of the next level ($l = 2$), $W^{(2)}_{\tau_{\beta-1},\tau_\beta}$. If the aggregated results are found to be

outliers against the density model of $W^{(2)}_{\tau_{\beta-1},\tau_\beta}$, then these samples are placed into the *outlier set* as potential clustered outliers. The aggregated results are also forwarded to the active window $W^{(2)}_{\tau_{\beta-1},\tau_\beta}$ in order to update its density model.

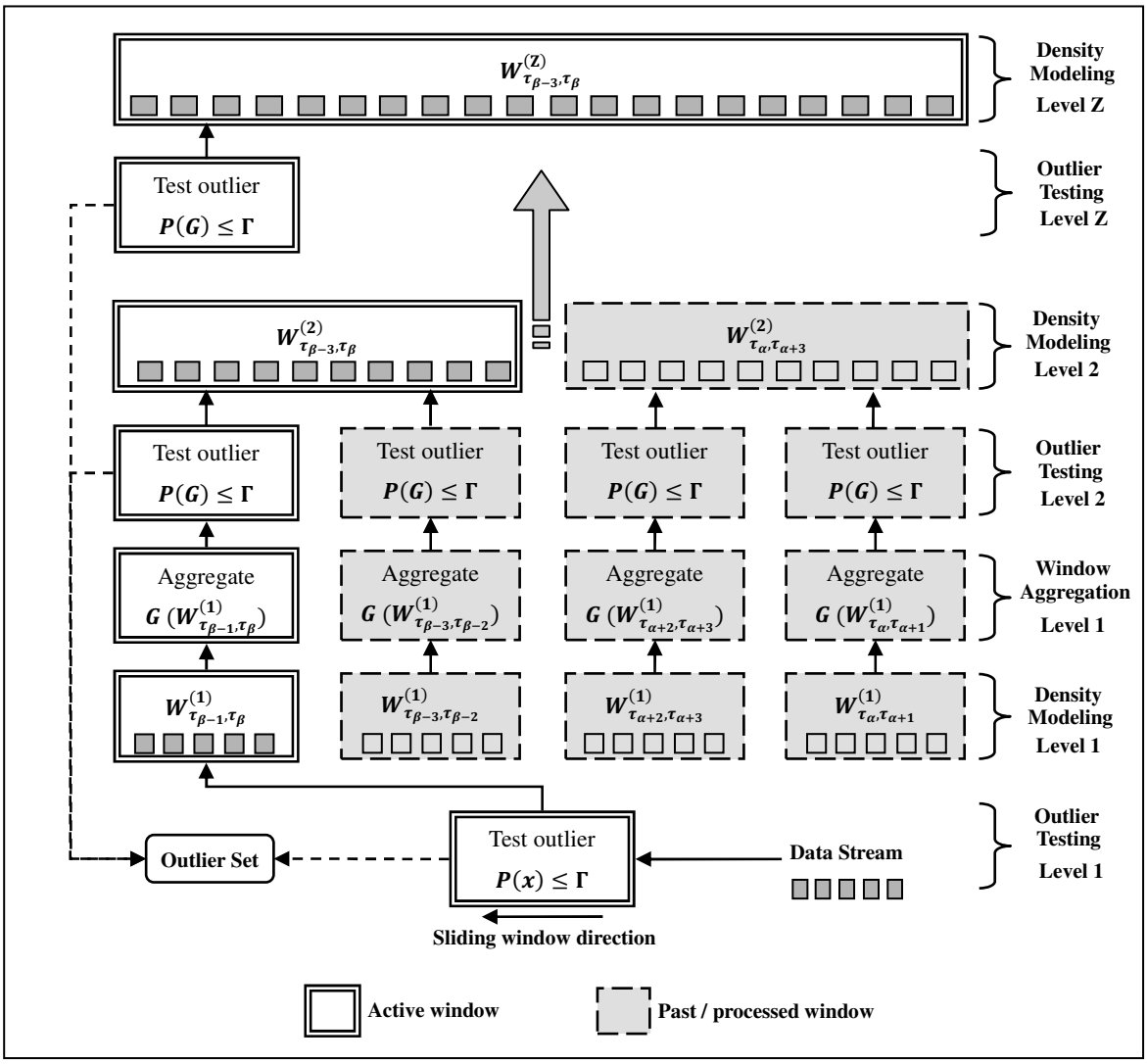


Figure 6.1 Multi-scale Outlier Detection Framework. Double solid lined windows are active windows/operations and dashed grey-filled windows are inactive past windows/operations.

6.1.3.2 Density Modeling

Assume that for a given active window at any level $0 < l \leq Z$, the probability density function (PDF) exists and that the generating process is stationary within the active window. Under these assumptions,

estimation of the PDF of any active window within hierarchy levels 1... Z can be effectively produced. The unknown and dynamic distributional structure of D implies that the PDF estimation method be non-parametric and can effectively adjust to concept drifts. Hence, the LR-KDE (see Chapter 3) is selected as the density estimation technique for this component. The multiple query optimizations of the LR-KDE is exploited to improve the throughput of query estimations. The LR-KDE also employs the time list structure which allows for effective and efficient maintenance of a time-based sliding window. Techniques such as M-Kernel [98] and Cluster Kernels KDE [41, 43, 46] cannot be directly transformed into a time-based window since the estimators employ the fading window model. Deriving the transformation function for a time-based window to a fading window model is difficult since the fading function is always applied to the entire sample set and does not maintain individual kernel fading weight/time stamps. However, in this work, we enhance the Cluster Kernels KDE by combining and utilizing LR-KDE's time list data structure to support the time-based window. Lastly, to increase the robustness of the original LR-KDE, the local region construction and regularization methods from GLEAM (see Chapter 5) are integrated into the LR-KDE. The regularization on local regions allows the LR-KDE to better adapt to data sets of varying complexity.

6.1.3.3 Window Aggregation

The objective of the window aggregation is to summarize a window W^l into a set of objects in window W^{l+1}. The commonly used aggregation function in multi-scale modeling is the expectation function [59, 94, 95]:

Equation 6.1

$$G(W) = E[W]$$

For the LR-KDE, since each local region is a mode of the distribution (see Chapter 5), the expectation function is revised such that it is conditioned on the local region. Hence, the aggregation function is as follows:

Equation 6.2

$$G(W|LR_i) = E[W|LR_i]$$

where LR_i is local region i.

To obtain all of the aggregated sample objects, we "marginalize" over all LR_i via the union operator as follows:

Equation 6.3

$$G(W) = \bigcup_{all\ LR} E[W|LR_i]$$

The above aggregation operation preserves the trend of individual local regions and thus propagates the *dominant* density features to the upper hierarchies. The traditional operation as defined in Equation 6.1 can mask significant density structures (e.g., modes) and introduce artifacts which can degrade estimation

quality and reduce detection performance.

6.1.3.4 Outlier Testing

For each new incoming sample o_l in level l (i.e., raw data stream sample in $l = 1$ and aggregated sample in $l > 1$), it is assumed that o_l is generated by the same mechanism that produced the samples in $W^{(l)}$. Hence, the outlier condition of o_l can be defined as a function of its probability $P_{W^{(l)}}(o_l - \varepsilon \leq o_l \leq o_l + \varepsilon)$ within the active window $W^{(l)}$. Specifically, an outlier is defined as follows:

Equation 6.4

$$Outlier(o_l) = \begin{cases} 1, P_{W^{(l)}}(o_l - \varepsilon \leq o_l \leq o_l + \varepsilon) \ \leq \Gamma \\ 0, \text{otherwise} \end{cases}$$

Using the estimated PDF, the probability is computed as follows using a KDE $\hat{f}_{KDE}$:

Equation 6.5

$$P_{W^{(l)}}(o_l - \varepsilon \leq o_l \leq o_l + \varepsilon) = \int_{o_l-\varepsilon}^{o_l+\varepsilon} f(x)dx \cong \int_{o_l-\varepsilon}^{o_l+\varepsilon} \hat{f}_{LR\_KDE}(x)dx$$

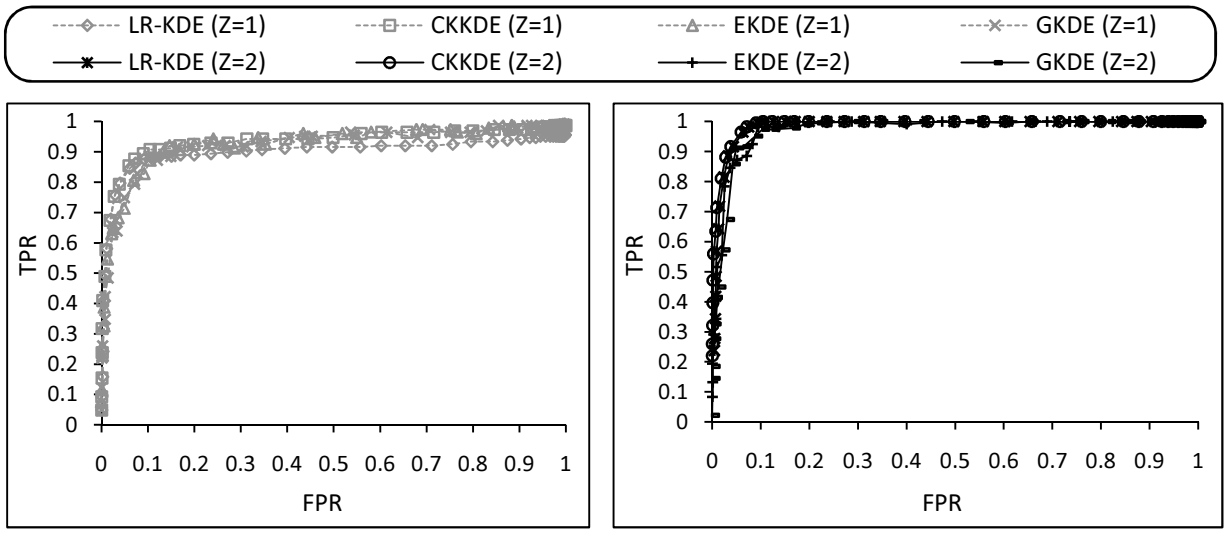
Hence, the probability $P_{W^{(l)}}(o_l - \varepsilon \leq o_l \leq o_l + \varepsilon)$ can be efficiently computed using the multiple query optimization technique proposed in Chapter 3.

6.1.4 Experiment

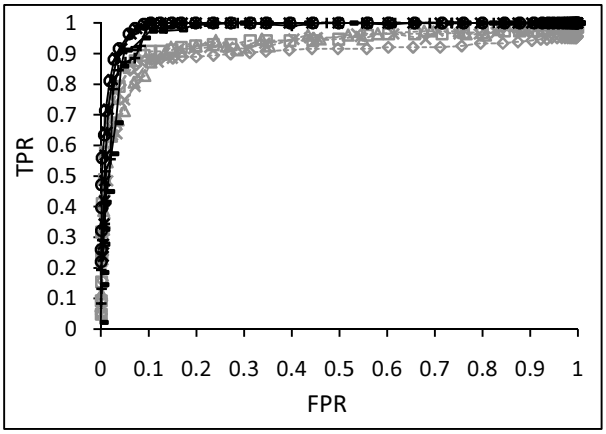
The following section provides the experimental results of the multi-scale outlier detection approach. The experiment focuses on the detection performance using the following metrics: true positive rate (TPR) and false positive rate (FPR). TPR is defined as the ratio of correctly detected outliers to the total number of outliers. FPR is defined as 1 – (the ratio of correctly classified non-outliers to the total number of non-outliers). Using the TPR and FPR values, the sensitivity plots or receiver operating characteristic (ROC) curves and areas under the ROC curves (AUC) are obtained. To test the impact of the hierarchical scheme, the outlier detection algorithm is executed for $l = 1,2$ where $l = 1$ represents the non-hierarchical approach. Furthermore, other existing KDE techniques are tested and compared within the proposed outlier detection framework. The existing set of KDEs include: Cluster-Kernels KDE (CKKDE), time sampling Epanechnikov KDE (EKDE), and time sampling Gaussian KDE (GKDE). For the data set, simulated time series based on MIX1, MIX2, MIX4, and MIX8 (see Appendix A) are employed. Randomly generated outliers derived from a mixture of normal composed of point and clustered outliers are injected into the data set. The parameter settings are as follows: Time length of $W^1 = 40$, time length of $W^2 = 200$, kernel size $M = 40$, and maximum local region size $Q_{MAX} = 10$.

6.1.4.1 MIX1 Results

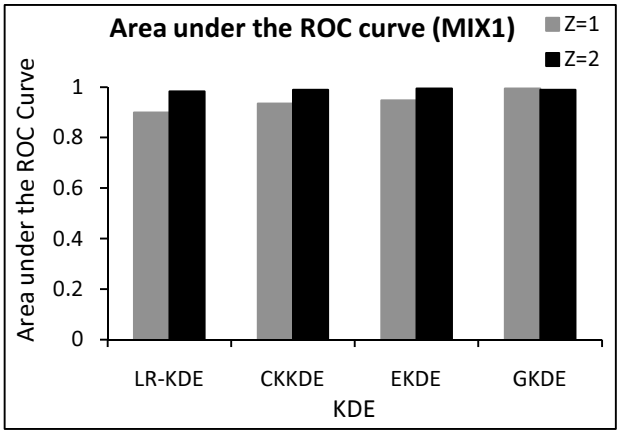
Figure 6.2 shows the results of the ROC curves and AUC for the MIX1 data set. For Figure 6.2(a)-(c) the *x*-axis is the FPR and *y*-axis the TPR. For Figure 6.2(d) the *x*-axis is the KDE technique with its corresponding hierarchical level and the *y*-axis is the AUC value. All of the KDEs exhibit high detection rates with the MIX1 data set which can be attributed to the simple and unimodal structure of the MIX1 distribution. For the non-hierarchical approach, the LR-KDE exhibits lower AUC than the other techniques. This result is due to LR-KDE's tendency to overfit simple distributional structure which can be addressed by imposing more aggressive regularization (see Section 5.2 for details). The results in Figure 6.2 also show that the detection performance of the LR-KDE is vastly improved when the hierarchies are employed. The increased improvement gives the LR-KDE the highest AUC compared to all of the non-hierarchical methods and remains competitive against the remaining hiearchical implementations. The hiearchical approach is also able to improve the detection performance of other KDE techniques: EKDE and CKKDE. The general improvement of the hierarchical approach is a result of the hierarchical method's superior ability to detect clustered outliers.



(a) ROC curves of $Z = 1$ (b) ROC curves of $Z = 2$



(c) Combined ROC curves of $Z = 1$ and $Z = 2$

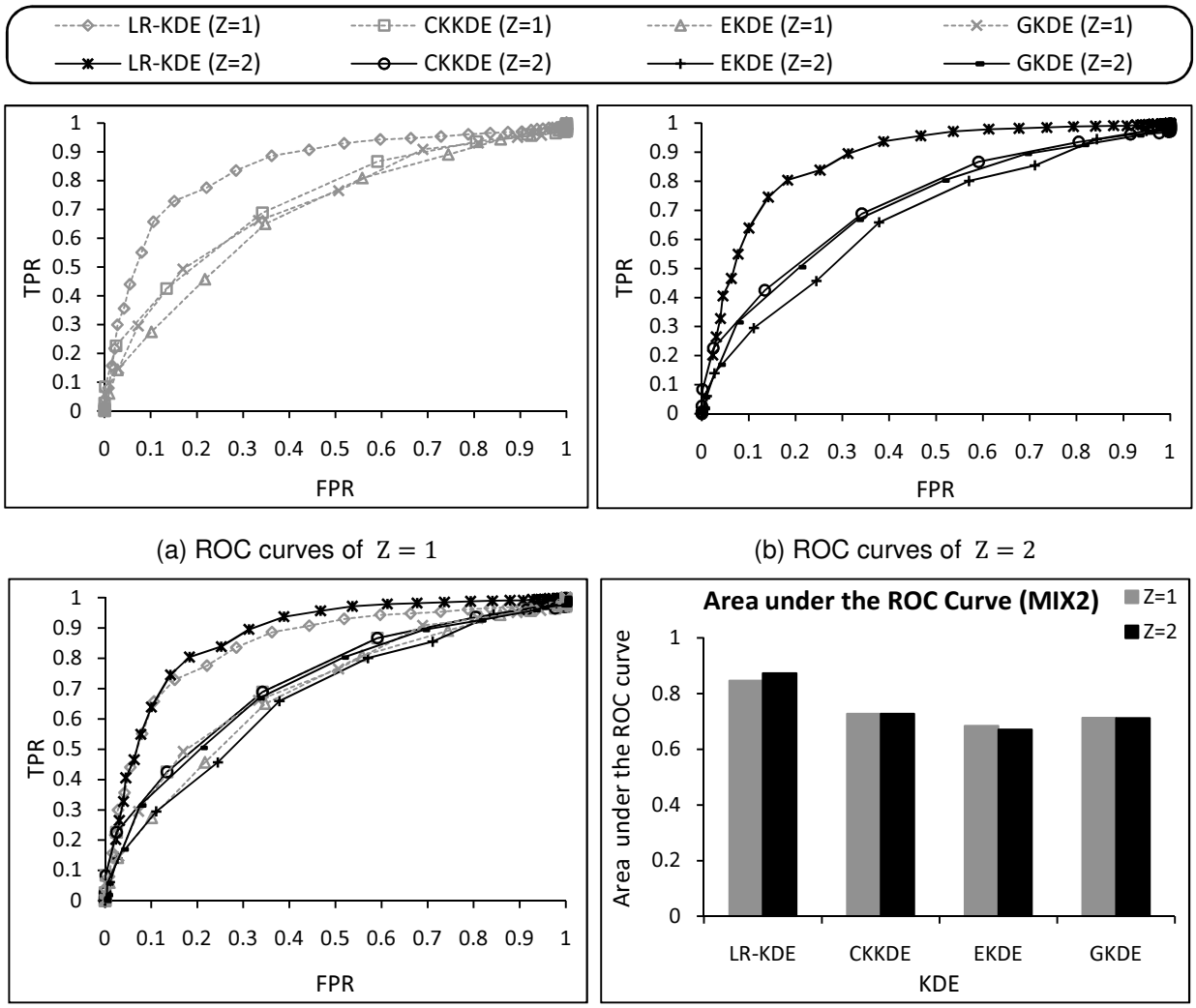


(d) Area under the ROC curve

Figure 6.2 ROC curves and areas under the ROC curves of MIX1 data where Z is the maximum hierarchical level

6.1.4.2 MIX2 Results

The results of the MIX2 data set are shown in Figure 6.3. For Figure 6.3(a)-(c) the *x*-axis is the FPR and *y*-axis the TPR. For Figure 6.3(d) the *x*-axis is the KDE technique with its corresponding hierarchical level and the *y*-axis is the AUC value. MIX2 provides greater composition sophistication than MIX1 as it exhibits two distinct modes. The increased complexity of the distribution is evident in the lower detection performance (compared to MIX1) of all the KDE techniques. However in MIX2, the LR-KDE consistently outperforms all other techniques due to its superior ability to model multi-modal features. Similar to MIX1, imposing the hierarchies improves the performance of the LR-KDE. However, invoking the hierarchies for other KDEs does not necessarily give additional improvements.



(a) ROC curves of $Z = 1$

(b) ROC curves of $Z = 2$

(c) Combined ROC curves of $Z = 1$ and $Z = 2$

(d) Area under the ROC curve

Figure 6.3 ROC curves and areas under the ROC curves of MIX2 data where Z is the maximum hierarchical level

6.1.4.3 MIX4 Results

Simulations results of the MIX4 data set are shown in Figure 6.4. For Figure 6.4(a)-(c) the *x*-axis is the FPR and *y*-axis the TPR. For Figure 6.4(d) the *x*-axis is the KDE technique with its corresponding hierarchical level and the *y*-axis is the AUC value. Similar to MIX2, the LR-KDE based outlier detection approach provides the highest detection performance in both the non-hierarchical and hierarchical classes. Like all the previous data set, invoking the hierarchies further increases the AUC of the LR-KDE. The lift in the ROC curve around FPR = 0.15 of the LR-KDE shows that the additional hierarchy improves the detection of clustered outliers.

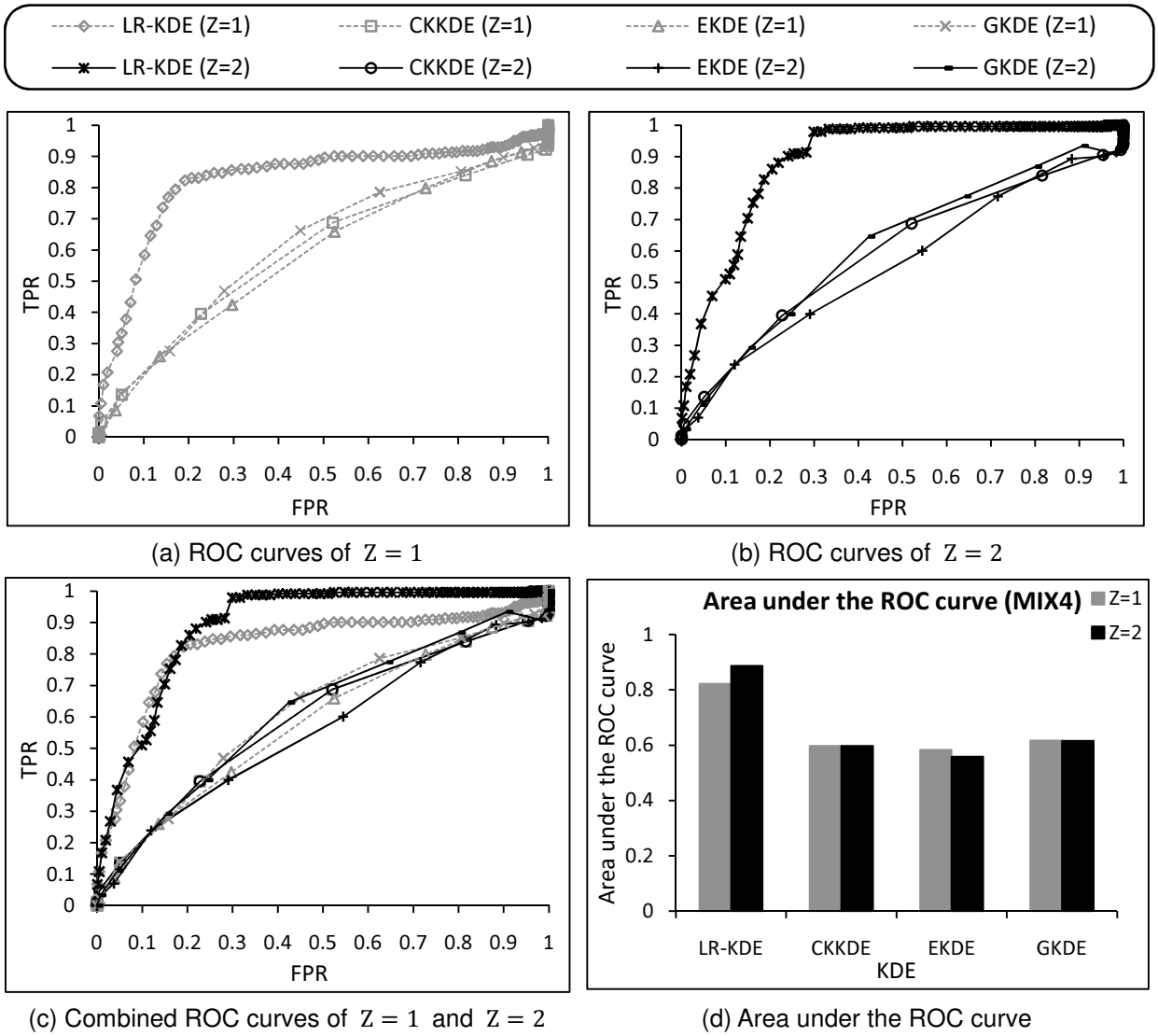
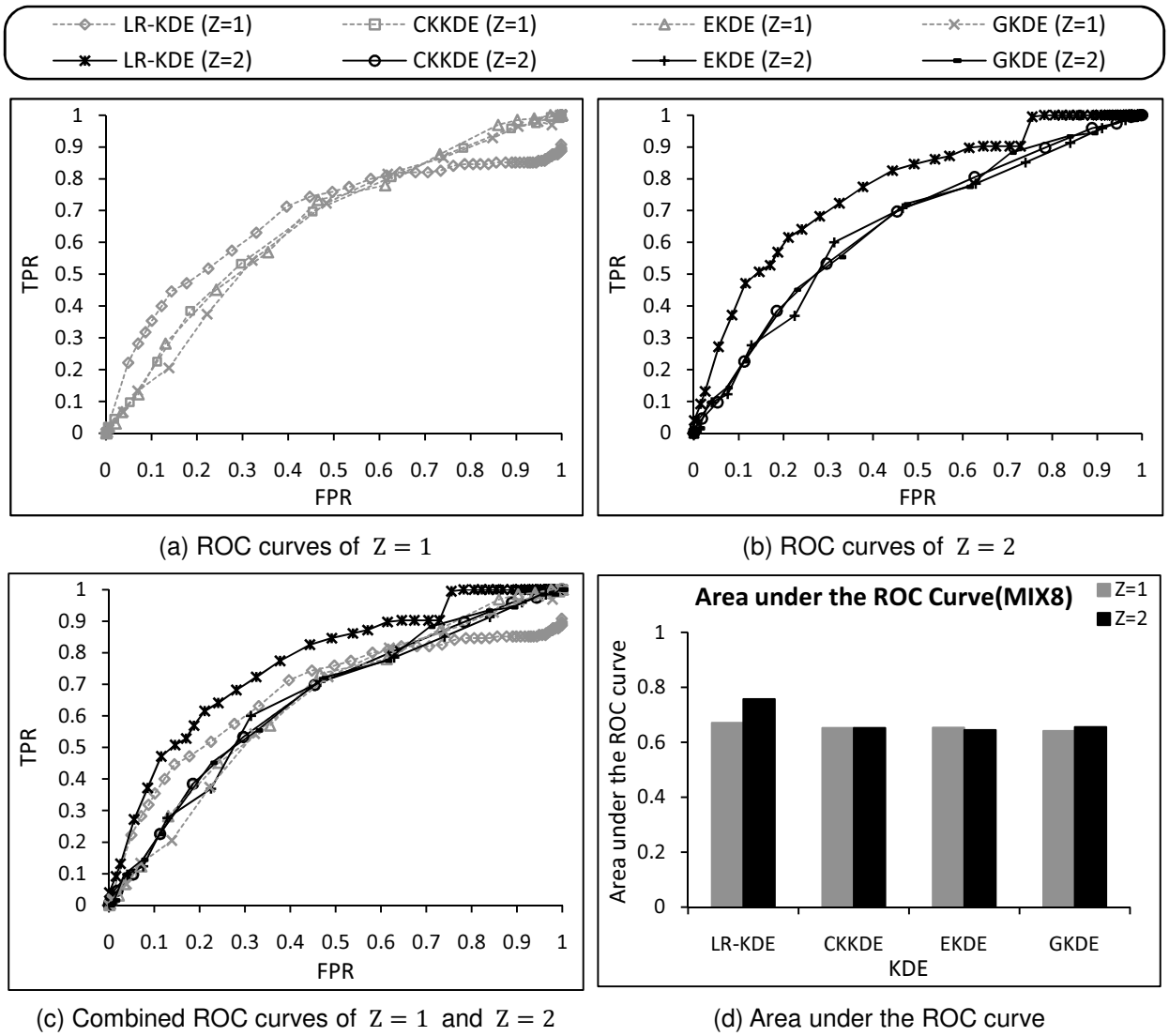


Figure 6.4 ROC curves and areas under the ROC curves of MIX4 data where Z is the maximum hierarchical level

6.1.4.4 MIX8 Results

The results of the MIX8 data set are provided in Figure 6.5. For Figure 6.5(a)-(c) the *x*-axis is the FPR and *y*-axis the TPR. For Figure 6.5(d) the *x*-axis is the KDE technique with its corresponding hierarchical level and the *y*-axis is the AUC value. For the MIX8 data set, the LR-KDE based method outperforms all other KDE techniques. Although the AUC of the LR-KDE for a given hierarchical level is higher than the competing KDE techniques, the AUC portion of the non-hierarchical LR-KDE between FPR = 0.7 to 1.0 is lower than the others. This occurrence is attributed to the LR-KDE's higher capacity (hence increased sensitivity) to capture distributional changes. Due to the influx of clustered outliers, the background

pattern at the finest scale window essentially adapts to the distributional pattern of the clustered outliers. Because the LR-KDE is able to quickly adapt to this sudden change, the clustered outliers could not be effectively detected by the non-hierarchical LR-KDE. Other non-hierarchical KDEs are significantly affected by this change due to their slower adaptation rate. However, when the LR-KDE is coupled with the hierarchical approach, the clustered outliers can now be detected by the high level density model. The combination of LR-KDE and hierarchies results in the best TPR values throughout entire the FPR range and thus the highest AUC amongst all other techniques.



(a) ROC curves of $Z = 1$

(b) ROC curves of $Z = 2$

(c) Combined ROC curves of $Z = 1$ and $Z = 2$

(d) Area under the ROC curve

Figure 6.5 ROC curves and areas under the ROC curves of MIX8 data where Z is the maximum hierarchical level

6.1.5 Conclusion

In this work, a multi-scale outlier detection framework is proposed which discovers both point and clustered outliers via a unified testing methodology and supports the use of an arbitrary density estimation technique. Furthermore, the framework is enhanced by integrating the LR-KDE which provides superior detection performance to existing KDEs such as CKKDE, EKDE, and GKDE. Empirical results show that the hierarchical approach consistently improves the detection rates of the non-hierarchical method when used in tandem with LR-KDE. Furthermore, coupling the LR-KDE with the hierarchical approach provides the best detection rates in almost all (3 of 4) of the tested data set.

6.2 Application II: Real-time Traffic Incident Detection

This subsection describes a set of approaches for the automatic detection of roadway traffic incidents and traffic prediction. Current automated incident detection techniques do not perform well under changing traffic patterns, recurrent congestions, and can require large amounts of training data. This chapter proposes a solution to mitigate these shortcomings by utilizing predicted traffic models and performing comparative analysis against observed traffic patterns to automatically detect incidents. The traffic models are summarized by way of descriptive statistics of the density function. This chapter is organized in the following manner. Section 6.2.1 provides the background and motivation. Section 6.2.2 summarizes the contribution of the work. Section 6.2.3 gives a detailed description of the proposed Adaptive Online Incident Detection (AOID) approach. Section 6.2.4 describes the case study performed on AOID. Lastly, Section 6.2.5 provides our conclusion.

6.2.1 Motivation

The provision of any emergency management system with respect to the public safety necessitates the inclusion of the transportation road network. Obstructions to traffic flow (i.e., congestions) can enormously reduce a region's ability to transport inhabitants and resources. This hindrance debilitates the region from invoking optimal response strategies to emergent events. The transportation network provides a means for mitigation plans (e.g., evacuation) for any disaster, whether it is natural or human-induced. If the transportation network is compromised, then the safety of the public infrastructure can be fatally handicapped. Therefore, it is critical that a region provide supporting infrastructures to protect their transportation road network.

The effects of road congestions not only impact the region's infrastructure but can severely hamper its economy and environment. It is estimated in 2005 that both recurrent and non-recurrent congestions cost $75 billion and wasted approximately 8.4 billion gallons of fuel [3]. The Texas Transportation Institute performed a nationwide study of 85 urban areas and reported that on average, an urban region (defined as a medium sized area) will lose $418 million due to congestion associated costs and waste 15 million

gallons of fuel [1]. Major contributors of congestions are non-recurrent congestions, which account for 13%-30% of the total traffic delay [83]. Non-recurrent congestions are results of incidents such as vehicular collisions, construction and maintenance activities, inclement weather conditions, or other activities which reduce the roadway capacity.

Several Intelligent Transportation Systems (ITS) have been deployed that employ mechanisms to minimize the damages ensued by roadway incidents. An important step in containing and reducing the damages caused by incidents is to minimize the time needed to respond to the event. ITS can employ a monitoring tool to observe the behavior of traffic within a region and provide an Automatic Incident Detection (AID) scheme to alert traffic operation personnel of a potential incident event. AID schemes are pivotal in providing maximum effectiveness for emergency personnel to react to an incident by reducing the time to detect the event. For the victims of incidents, their chance of fatality rises 6% for every minute of delay it incurs on the emergency team's response time [28]. Furthermore, the chance of secondary incidents rises as the duration of an incident increases and travel times for commuters suffer due to incidents. A tool that ITS can leverage for performing road safety analysis is traffic trend prediction, which provide traffic planners and other key decision-makers insights into building effective roadway designs. These decisions can impact the ability of a roadway to handle sudden increased loads and moderate traffic capacity to reduce recurrent and non-recurrent congestions.

Much research and development have been emphasized on the subject of traffic incident detection. However, many current methods have drawbacks that can render them ineffective for the use in emergency management systems. Due to computational time constraints and their inability to automatically adapt to evolving traffic behaviors, the current set of AID may not be suitable for use in the unpredictable and highly dynamic setting of emergency management systems. Over the past two decades, several automatic incident detection methodologies have been proposed, such as [70, 72, 79, 92]. These methods are able to determine incidents with high detection rates but must be performed within strict environmental parameters, such as specific temporal constraints and minimum traffic volumes. When the road behavior changes, approaches such as these may become less effective and require manual re-adjustments of system parameters. Other methods that can work under changing road conditions are based on computational intelligence. Many of these schemes utilize neural networks, Bayesian networks, and fuzzy logic [4, 66, 84]. However, these techniques can require large training datasets (which may not be readily available) and in some cases incur high runtime cost that cannot meet the computational constraints of emergency management systems.

6.2.2 Contribution

The AID scheme developed in this chapter adapts to the varying dynamics of the environment, requires a small set of training data, and efficiently utilizes the system's computational resources. ***A***daptive ***O***nline ***I***ncident ***D***etection (AOID) is the proposed contribution to alleviate these deficiencies of current AID approaches. The AOID system is implemented as an extension to the ***A***dvanced ***I***nteractive ***T***raffic

Visualization ***S***ystem (AITVS) [58]. The AITVS, developed by Virginia Tech's Spatial Data Management Lab, is a comprehensive traffic visualization system that presents summarizations of spatiotemporal patterns of road detector data in the Metropolitan Washington D.C. areas. AITVS marries a wide set of the multidimensional visual components with efficient processing algorithms to deliver a responsive and comprehensive traffic visualization system. The AOID contributes and integrates two sets of tools into AITVS: recurrent traffic behavior forecasting and automatic incident detection. These tools supplement AITVS to provide emergency personnel the information required to quickly devise and invoke an emergency plan. Furthermore, the information derived from the tools can be interfaced with traveler information systems such as highway variable signs to give commuters the most up-to-date traffic conditions.

6.2.3 Proposed Approach: AOID

AOID is a unified solution that targets the inter-dependent requirements of incident detection and traffic forecast. Traffic incident can be qualitatively described as follows: a *traffic incident* exhibits as an anomaly that diverges for some threshold *m* (e.g., standard deviation) from the *forecasted traffic* value. Since only a subset of the anomalies is incident related, it is necessary to provide a classification mechanism to decipher which of the observed anomalies are representatives of actual incidents. The generated traffic trend model can adapt to evolving recurrent traffic patterns that are induced by changes in environmental parameters (e.g., long-term road construction, road expansions). Using the traffic trend models, the system determines the deviations of currently observed traffic values. The deviations are analyzed and processed by a nearest-neighbor classifier to detect incident occurrences. Hence, the architecture is divided into the following subtasks: **data preprocessing**, **model formulation**, **anomaly detection**, **classification**, and **data visualization**.

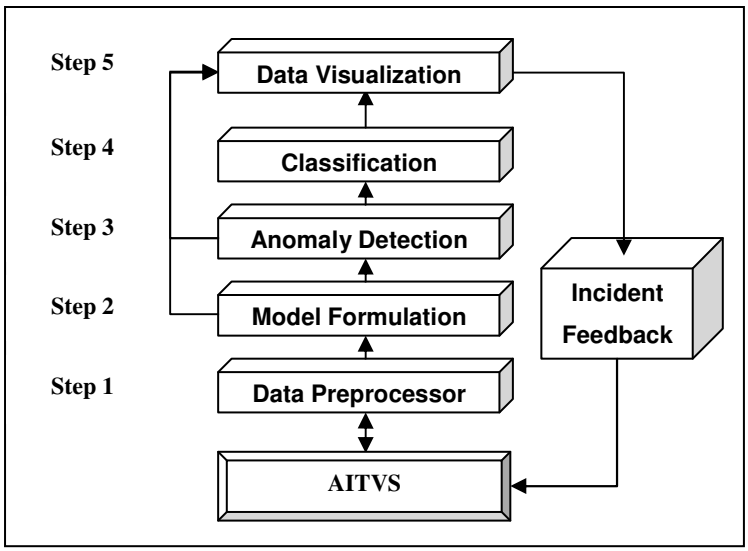


Figure 6.6 Architecture of AOID

In the AOID (see Figure 6.6), detection is performed as follows. Firstly, a traffic trend model is generated for the given day. Then that day's observed traffic value is compared to the generated model and determined if anomalies exist within that observation. If anomalies are found, then determinations of the anomalies' incident probability values are made. The AOID system is tightly coupled with the AITVS, and as such it inherits all the low-level data preprocessing and access functions of the AITVS. Raw sensor data arriving at the AITVS server are parsed and stored in the AITVS traffic database. AOID utilizes this database, but the data are not filtered for noises such as sensor fluctuations and missing values. Therefore, it is the task of the **data preprocessor** in the AOID to furnish usable and cleaned traffic data to the remaining AOID subsystems. From the data preprocessor, traffic data is passed to the **model formulation** module to generate the traffic models in real-time. Then current traffic observations are sent to the **anomaly detection** module along with the generated traffic models. If anomalies are found, then this anomaly dataset is sent to the **classification** module. The classification gives the incident probability of the anomaly set.

6.2.3.1 Data Preprocessor

To perform the mining tasks, historical traffic data will need to be cleaned for noise, marked for incident related data, and labeled for short-term traffic events. Data noises include malfunctioning sensors and missing values. Marked or labeled records will be used to differentiate data stemming from normal traffic activity. The data preprocessing task is outlined as follows:

1. **Noise filtration –** Scan the database for non-conforming records that are likely to originate from malfunctioning detectors or invalid values, and remove these records.

2. **Incident filtration –** Find all records that are known to be associated with incident events and mark this information. These events include vehicular collisions, highway maintenance, and other incidents reported by the DOT database.

3. **Short-term event filtration –** Find all records that are associated with holidays or special events, and mark this information.

4. **Output –** Furnish traffic records that are associated with normal and recurrent traffic behavior.

A rule-based algorithm is used to identify and remove data noise. Incident related data and short-term events will need to be correlated with the incident report and traffic event database, which are performed using database join routines. Join routines are costly functions that can take $O(|n| * \log |m|)$ time complexity where m and n are the datasets and a logarithmic search time complexity index (e.g., B^+-tree) applied to m. Hence, to improve the performance of the preprocessing step, an incremental approach is proposed. The AOID utilizes an agent that monitors newly arriving data in AITVS. The agent

processes the new data at designated intervals to perform the necessary filtration tasks. The processed data are then re-inserted into the AITVS database for use with the model formulation and anomaly detection tasks of AOID. Therefore, preprocessing only occurs with newly arrived data.

6.2.3.2 Model Formulation

This module generates weekly traffic trends (e.g., traffic models) for each day of the week. For each day, the model captures the same set of days of preceding weeks to generate the predicted trend. For example, to generate the traffic model for Monday of the 4th week of July, the module will obtain the traffic data from Monday of the 3rd, 2nd, and so forth of July and possibly extending to June. The number of weekly data is an adjustable parameter. After obtaining the past weekly data, the model will use a weighted average to produce the expected value (i.e., trend). Weighted average will allow the model to adapt to changes in traffic patterns. The generated models will be used to determine deviations in current traffic values. Depending on the scope of the analysis, the models can be directly sent to the to the **data visualization** module for further examination.

To describe the traffic trend generation algorithm, define the following parameters, functions, and operations:

- Let $\vec{A}$ and $\vec{B}$ be vectors of n components then define the following component-wise operations:
  - Vector multiplication: $\vec{A}\vec{B} = [\dots, a_i b_i, \dots]$.
  - Vector root: $\sqrt{\vec{A}} = [\dots, \sqrt{a_i}, \dots]$.
- Let $\vec{F}(s, w, d, t)$ be a function that gives the observed traffic vector on station s, week w, day d, and time step t.
- Let $V = \begin{bmatrix} F(s, w_1, d, t) \\ \vdots \\ F(s, w_n, d, t) \end{bmatrix}$ be a matrix where w_i is the week prior to w_{i-1} and n is the number of weeks. For ease of notation, $V[i]$ will refer to the ith row of V.
- Let α_i be the scalar weight given to each component of V.
- Define an adjustable inter-weight relationship θ.

The predicted value is defined as follows:

$$\vec{F}\left(s, w_{predicted}, d, t\right) = \sum_{i=1}^{n} \alpha_i V[i]$$

where α_i is conditioned as follows:

- $\alpha_i > \cdots > \alpha_{i+1} > \cdots > 0$ i.e., higher weights are assigned to more recent data.

- Unity property:

Equation 6.6

$$\sum_{i=1}^{n} \alpha_i = 1$$

Because the model adapts to changes in traffic behavior, it will necessarily assign higher weights to more recent data. For a given set of β_i's that satisfy the unity constraint (Equation 6.6), define α_i as follows:

Equation 6.7

$$\alpha_i = [\sum_{i=1}^{n} \beta_i]^{-1} \beta_i$$

The above equation expresses the unity constraint in a more intuitive form and eases the search for formulae that will satisfy the requirements.

For this work, a linearly proportional and gradual weighting formula is employed. Define a parameter k such that k is a scalar multiple of the oldest datum weight and that k will be the initial value assigned to the most recent datum weight. Hence, the relationship of β_i and its older neighbor can be described as follows:

$$\beta_i - \beta_{i+1} = \frac{k}{n-1}$$

Removing the recurrence relationship of the above equation gives the following:

Equation 6.8

$$\beta_i = \beta_n + (n - i)\frac{k}{n-1}$$

Let $\beta_n = 1$ and derive the closed-form summation formula for β_i as follows:

Equation 6.9

$$\sum_{i=1}^{n} \beta_i = \frac{n(kn+n-k-1)}{2n-2}$$

Using equations Equation 6.7, Equation 6.8, and Equation 6.9, the following final closed form solution for the linear weights α_i is derived:

$$\alpha_i = \left(\frac{2n-2}{n(kn+n-k-1)}\right)\left(1 + (n-1)\frac{k-1}{n-1}\right)$$

If $k = \theta$ then k can be established as the learning rate parameter for the above weighting formula.

6.2.3.3 Anomaly Detection

The task of this module is to determine anomalies for the currently observed values. Along with the traffic

trend model, the deviation of the traffic model is also provided by the **model formulation** step. The deviation will provide the threshold for classification of normal traffic values. Otherwise, if the new value at that time slot differs more than the threshold then this new value is considered as an anomaly. The standard deviation vector is calculated as follows:

$$\overrightarrow{SD} = \frac{1}{n}\sqrt{\sum_{i=1}^{n}\left(\alpha_i\left(V[i] - \vec{F}\left(s, w_{predicted}, d, t\right)\right)^2\right)}$$

The expression above involves the component-wise vector operations defined in Section 6.2.3.2. Now, define vector $\vec{D}$ as the component-wise L_1 norm of the predicted traffic vector and current traffic:

$$\vec{D} = |\vec{F}\left(s, w_{predicted}, d, t\right) - \vec{F}(s, w_{current}, d, t)|$$

The current observation, $\vec{F}(s, w_{current}, d, t)$, is defined to be an outlier if any component of $\vec{D}$ exceeds its corresponding component in $\overrightarrow{SD}$.

This task highlights the importance of the preprocessing stage. Because the calculation of $\overrightarrow{SD}$ and $\vec{D}$ are sensitive to perturbations, it is critical that the data used in the model generation are generally free from noise and known outliers (e.g., incidents). It is also worthy to note that the detection of an outlier follows a straightforward $O(C)$ operations where C is the number of components in $\vec{D}$. Recall that $\overrightarrow{SD}$ is derived in the model formulation step hence not part of the online processing. The efficient $O(C)$ implementation of the outlier detection serves as a filtering step to the potentially expensive process of incident classification. The use of the *filter* and *refine* approach within the context of AOID can improve overall detection performance.

6.2.3.4 Incident Classification

This module verifies that the anomalous data reflect previously known incidents. Define an *incident class vector* $\vec{I}$ as the descriptive structure that model a particular class of incidents. Specifically, the components of $\vec{I}$ are the expected deviations of the traffic vectors of known incidents: occupancy, speed, and volume. This scheme gives AOID the ability to classify incidents into various types. For example, the incident vector for inclement weather will be markedly different from the incident vector associated with vehicular collisions. The classification scheme afforded by this component is essentially the nearest-neighbor classification strategy.

Define the incident probability measure P as follows:

$$P = 1 - \frac{Distance(\vec{D}, \vec{I})}{Distance(\vec{F}\left(s, w_{predicted}, d, t\right), \vec{I})}$$

The above incident probability formulation allows a generalization to a variety of distance metrics. For example, the Mahalanobis distance may be used as a measure. Mahalonobis distance requires the computation of the inverse covariance matrix which can incur a cost that is quadratic to the number of dimensions. However, because much of the traffic streams have been filtered for anomalies, the number of tested samples processed at this stage has been drastically reduced.

There exist some practical settings for which the single-loop detector data is the only available source of traffic information. The single-loop detector provides the three aforementioned traffic attributes, however, only the occupancy attribute is provided in its raw observed form. The other two components (speed and volume) are obtained as linear derivation of the occupancy value. In such a case, employing the Euclidean distance measure on the occupancy attribute can provide for a highly efficient and effective metric.

Because traffic behaviors evolve over time, incident vectors can become stale and will need to be removed from the system. To update the set of incident vectors, AOID employs the following heuristic: delete the incident vector if all of its components are within the support of $\overrightarrow{SD}$. The provision of this heuristic is to retain semantic consistency with the employed outlier measure by enforcing the subset relationship between incidents and outliers.

6.2.3.5 Data Visualization

This module serves as an interface to the user and various subsystems of AOID. The module creates a graphical representation from the following components: model formulation, anomaly detection, and classification. For the model formulation, volume, speed, and occupancy plots for both the currently observed traffic patterns and the generated traffic models are shown (Figure 6.7(a)). Lastly, incident probabilities of the currently observed values will be plotted on a time series graph (Figure 6.7 (b)).

6.2.3.6 Real-time Application Use

Adapting AOID for real-time environment requires that the system only provide online processing for those tasks that make use of the current data stream and delegate the other set of tasks to offline processing. Because **anomaly detection**, **incident classification**, and **incident feedback** can be effectively done online, these components will not require any algorithmic or design alterations for applications in a real-time scenario. However, the **model formulation** will need to be re-adapted to serve as a summarization (synopsis) structure for which to compare current arriving data. Since the model formulation only requires information from *past records* in the database, it can be generated before the real-time monitoring task is invoked. For example, real-time monitoring for i^{th} day will utilize the summarization structure that has been pre-generated at some previous j^{th} day (i.e., $j < i$). Therefore, by invoking the model formulation task as an off-line process, the AOID can serve as a real-time incident

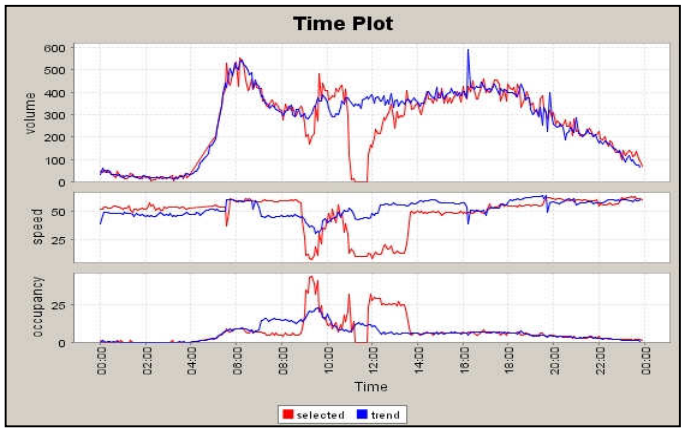
monitoring system.

6.2.4 Case Studies

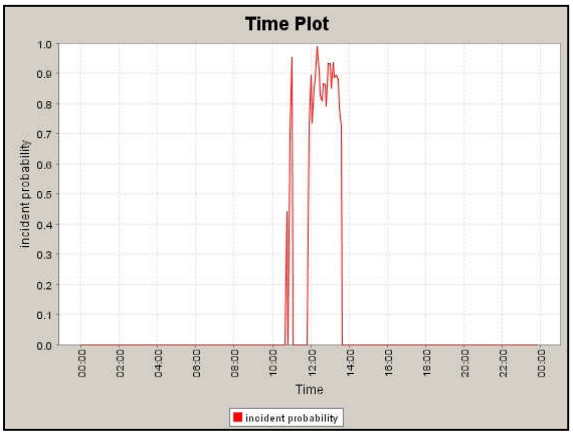
The AOID is implemented using Java 1.4.2 and designed as a subsystem of AITVS. AITVS provides interfaces for traffic data access and visualization. Low-level data processing is performed by the AITVS. Tasks such as data fusion and translation are dispatched within the AITVS to a standard format that can be efficiently accessed by the AITVS subsystems. The case study use incident cases of I-66 from January 2004 to July 2005. The following provides in-depth discussions of three traffic events: one on 5/3/2005 at station 331 (event 1), second on 7/24/2005 at station 341 (event 2), and third on 5/1/2005 at station 261 (event 3).

6.2.4.1 Traffic Case: Event 1

For event 1 (Figure 6.7), an incident occurred between 10:15AM to 10:30PM with a peak probability of approximately 0.95. The incident is verified via the Virginia Department of Transportation (VDOT) incident database. For this event, VDOT reported that a vehicular collision occurred at 10:16AM and cleared at 10:27AM which coincides with the first spike in the incident graph. This spike is indicative of the initial impact of a vehicular collision. At initial impact, immediate trailing vehicles will reduce their speeds dramatically and form an anomalous cluster dataset which translates to the first spike of the incident probability. However, oncoming traffic will decrease its speed (regardless of the fact that the incident has been cleared) due to debris or other slow moving vehicles which accounts for the presence of the second and following spike. The second spike is not directly indicative of a separate incident but rather a subsequent congestion (i.e., secondary incident) that resulted from the first incident. Because this is a non-recurrent congestion, the AOID will regard this as an incident and could potentially cause a false alarm.



(a) Traffic plot

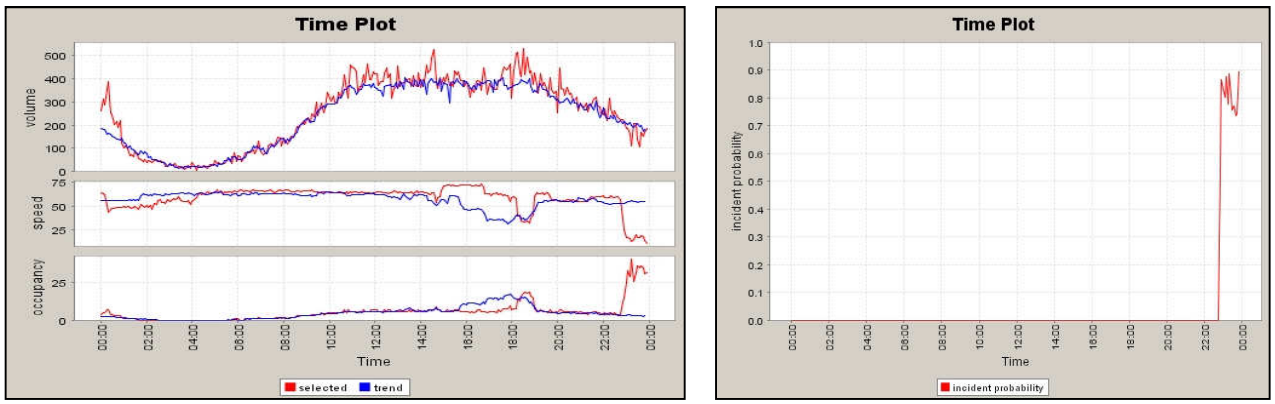


(b) Probability plot

Figure 6.7 Traffic and incident plots for event 1 (5/3/2005, EB, 331)

6.2.4.2 Traffic Case: Event 2

In event 2 (Figure 6.8), the effectiveness of the AOID is demonstrated by identifying incidents in the presence of non-incident anomalous data. Between 3:30PM and 6:30PM there is a divergence in traffic behavior as occupancy is lower than expected. The AOID views this event as an anomaly but is dismissed at the classification stage as it does not exhibit the behaviors of an incident. However, at approximately 10:30PM during low volumes, the AOID correctly shows that there is a 0.85 probability that an incident occurred.



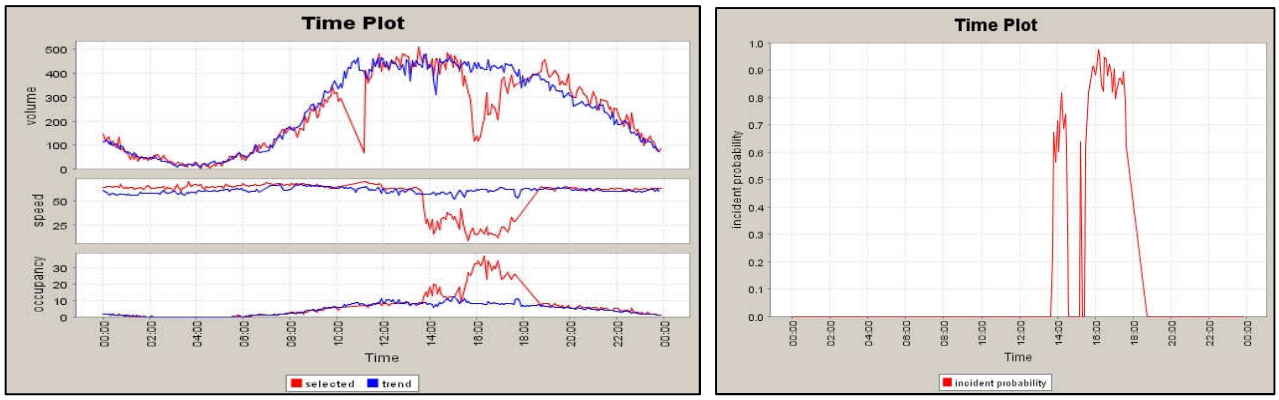
(a) Traffic plot (b) Probability plot

Figure 6.8 Traffic and incident plots for event 2 (7/24/2005, EB, 341)

6.2.4.3 Traffic Case: Event 3

In event 3 (Figure 6.9), it is observed that an incident occurred at around 1:30PM-7:00PM. Similar to the above events, the duration of the incident for event 3 can also be evaluated by estimating the length of the incident probability graph for those values that exceed an incident probability threshold. The gap between 2:30PM-3:00PM is indicative of the situation that is explained in event 1. Prior to the incident, at approximately 10:00AM to 11:45AM there is a dramatic drop on the volume curve due to a malfunctioning detector. Much of this data is detected and filtered at the preprocessing stage. However, it's neighboring data anomalies (Figure 6.9(a)) at 9:30AM-10:00AM and 11:45PM-12:00PM, have not been removed but have been transferred onto the remaining AOID components. At the classification stage, these anomalous datasets are assigned low similarity scores due to their large distances from the incident vectors; hence, they are categorized as non-incidents. This demonstrates a critical step in removing all such potential false alarms.

Eight actual incident cases were used to evaluate the AOID and in each case the AOID was able to detect the incidents with probabilities higher than 0.70. Due to the unavailable incident cases, detection rate and false alarm rate metrics were not considered. However, the chapter provides a set of case studies covering a wide scope of incident types to observe and validate AOID's approach.



(a) Traffic plot (b) Probability plot

Figure 6.9 Traffic and incident plots for event 3 (5/1/2005, EB, 261)

6.2.5 Conclusion

In this chapter, two sets of traffic analysis and monitoring tools were proposed: traffic trend prediction and automatic incident detection. The tools provide a framework for which application specific knowledge (i.e., learning rates and outlier distance metric) can be integrated to effectively detect incidents. The implemented AOID utilized a linearly proportional weight learning scheme to generate the traffic models and predictions for VDOT data. In summary, the AOID system detects incidents within various and dynamic traffic environments, requires minimal training dataset, and efficiently uses the system's computational resources.

Chapter 7. Research Contributions and Future Work

This research centers on the development of efficient and effective probability density estimators for data streams. This work proposes the concept of local regions to improve the KDE's estimation quality for complex data streams. Three research areas have been outlined for the local region based KDE which includes: theoretical development, algorithmic design, and application. Results have shown that the local region approach provides enhanced estimation quality over existing stream-based KDEs through the demonstration of a specific algorithmic implementation, LR-KDE. Furthermore, key theoretical properties of the local region approach are derived and extensively analyzed which include its consistency, convergence rate, and error reduction capacity. Based on the results of the analyses, the generalized local region algorithmic framework (GLEAM) is proposed and applied to existing stream-based KDEs. Comprehensive experiments have shown that GLEAM can effectively and significantly reduce the estimation errors of existing techniques while retaining a linear order runtime. Optimization techniques are constructed to further improve the runtime efficiency of GLEAM. This work also includes the design of a fast multiple density query approach to allow efficient processing of data streams to accelerate tasks such as visualization and outlier detection. In the application arena, this research applied the Local Region based KDE to the problem of multi-scale outlier detection and developed methods based on PDF statistics to automatically detect incidents in highway data.

7.1 Research Achievements

The three core research areas define the following major research tasks in this thesis: (1) theoretical development of the local region based KDE; (2) algorithmic design of the local region KDE; and (3) application of the local region KDE. The following provides a summary of the achievements of this research work:

7.1.1 Theoretical Development

Local Region Concept

The concept of the local region is proposed to approximate the adaptive kernel density estimator. The local region provides a piece-wise bandwidth assignment that considers the local features of the distribution. To effectively isolate the local features, the pair-wise adjacent theorem is proposed. The pair-wise adjacent theorem also supports an efficient algorithmic implementation suited for the data stream environment. Consistency of the estimator is demonstrated through the use of Parzen's sufficiency condition. Due to the large sample sizes afforded by data streams, consistent estimators can exploit this data stream property to further reduce estimation errors. Consistency of the local region KDE was also supported by the empirical studies where the RMSE is reduced as the number processed samples is increased.

Expected Performance

The bias, variance, mean squared error, mean integrated squared error, and asymptotic mean integrated squared error were derived and analyzed for the local region based KDE. The derivation of these error forms allowed analysis of the local region based KDE's convergence rate and its performance under practical scenarios. The error measures were also evaluated against the standard KDE, which represents the current best performing class of estimators for data streams.

Error Reducing Condition

The condition by which the local region based KDE can ascertain lower estimation error than the standard KDE is obtained. The result shows that in the general case, lower expected estimation error is attained when the parametric difference ratio is lower than the aggregated curvature. By utilizing appropriate local region construction methods and applying it to structurally complex distributions, a relatively low parametric difference ratio can be obtained to generate high quality estimates. Extensive empirical evaluation on the improvement of the local region based KDE has been conducted under various dataset conditions including those with concept drifts.

Local Region Construction

The asymptotic mean integrated squared error of the local region KDE under the LR-compatible Scott's Rule bandwidth shows that the local regions can be construed as isolating partitions of the distributional modes. Furthermore, if the modes are allowed to be estimated by a mixture of Gaussians, then existing clustering techniques can be applied to construct the local regions. One such technique is the K-means which possesses variants to address the problems of data streams. A regularization algorithm is developed that can automatically determine the number of true modes within the distribution. Experimental results show that the construction scheme is able to provide high quality estimates compared to existing techniques.

Sufficiency Condition for Local Regions

The prerequisites for reducing the expected error under the application of local regions are analyzed. The local region based KDE presumes that the samples are independent and identically distributed. This assumption guarantees that estimation is performed on a sample set that is identically distributed to the original set. Several quantized representation scheme exists (e.g., cluster-based, grid-based) to approximate the original dataset. If these approximating strategies sufficiently model the original distribution, then applying the local region to existing KDEs can enhance their estimation quality. Experimental results show that the application of local regions to existing KDEs reduces the estimation errors.

7.1.2 Algorithmic Design

Local Region KDE Technique

The local region KDE (LR-KDE) technique is proposed to estimate complex densities under a sliding

window model. An algorithm to incrementally construct local regions (utilizing the pair-wise distance theorem) is developed that can efficiently operate within the parameters of the data stream environment. Sample set representation is achieved by employing a clustering based quantization of the data samples. The developed sliding window model supports a time-based weighting strategy to update the clustered samples. The asymptotic time/space complexities are analyzed and the results indicate that the LR-KDE satisfies the requirements of the data stream environment. Empirically, the LR-KDE has shown that it provides higher estimation quality than existing approaches and improves update and query performances.

Generalized Algorithmic Framework

Due to the theoretical results of the sufficiency conditions of local region based KDE and the positive results from initial empirical validation, the Generalized Local rEgion AlgorithM framework is proposed. GLEAM provides efficient construction and query processing algorithms to support the integration of existing stream-based KDEs. GLEAM employs both incremental (local region seeding) and on-demand construction (local region regularization) methods to effectively generate the local regions. Experimental results demonstrate that the GLEAM approach improves the estimation quality of existing stream-based KDEs and can provide significant estimation enhancement for complex distributions.

GLEAM Query Optimizations

Two optimization techniques (*heap-based regularization* and *hybrid kernel aggregation and filtering*) are proposed to improve GLEAM's query processing throughput. These optimization strategies enhance the performance of the local region regularization operation and reduce computation of the kernel objects' density contributions. Comprehensive experiments show that the optimization strategies improve the query throughput of GLEAM and in some cases can produce faster performance than the baseline (i.e., non-GLEAM) approaches. The query optimizations allow GLEAM to effectively subsume the crucial characteristics of the underlying KDE while minimizing the impact on query throughput.

Multiple Queries and Range Query Optimizations

Algorithms for optimal processing of multiple density queries and range queries are proposed. The proposed approach will follow two venues: traditional solutions and approximate solutions. Traditional solutions will focus on analyzing the minimal operations required for producing the best achievable estimates for the given sample set and bandwidth function. The approximate solutions provide an estimate of the traditional solutions. An example of the approximate solution is the spline-based regression of the kernel density estimates.

7.1.3 Application

Real-time Traffic Incident Detection System

An automatic traffic incident detection system is designed as a part of the AITVS, a real-time highway monitoring tool. The incident detection approach employs an adaptive model (based on PDF statistics)

that captures traffic evolution that allows accurate detection under changing traffic behavior. To improve its runtime, incident detection is performed through a filter and refine paradigm. Firstly, traffic anomalies are detected via simple and fast operations. The detected anomalies are then evaluated by an incident classifier which can employ more sophisticated algorithms. The proposed system is tested under a number of incident cases and results show that it can successfully detect incidents.

Data Stream Monitoring

The local region based KDE is applied to the task of data stream monitoring. Specifically, this component focuses on developing an approach for online multi-scale outlier detection. For this task, the PDF estimates are queried in multiple sets and processing these queries under the existing techniques is inefficient due to redundant computations performed on the local regions. In this research component, the multiple queries and range query optimizations are utilized to support efficient processing of real-time data stream monitoring.

7.2 Publications and Accomplishments

In Progress

Arnold P. Boedihardjo, Chang-Tien Lu, Feng Chen, Jing Dai: "Multi-Scale Outlier Detection for Data Streams", SIAM Data Mining, 2010

Arnold P. Boedihardjo, Chang-Tien Lu: "Hierarchical approach to Spatial Indexing on B-Trees", International Journal on Advances of Computer Science for Geographic Information System (Geoinformatica), 2010

Under Review

Arnold P. Boedihardjo, Chang-Tien Lu: "Fast Adaptive Kernel Density Estimation on Data Streams", IEEE Transactions on Knowledge and Data Engineering, March 2010

Arnold P. Boedihardjo, Chang-Tien Lu: "A Framework for Exploiting Local Information to Enhance Nonparametric Density Estimators in Data Streams", IEEE Transactions on Knowledge and Data Engineering, January 2010

Published

Feng Chen, Chang-Tien Lu, Arnold P. Boedihardjo: "GLS-SOD: A Generalized Local Statistical Approach for Spatial Outlier Detection", in Proceedings of the 16th ACM Conference on Knowledge Discovery and Data Mining, July 25-28, 2010

Arnold P. Boedihardjo, Chang-Tien Lu, James A. Shine, Michael Tischler: "Fundamental Algorithms for Mining Data Streams", in Proceedings of the USACE Research and Development Conference, Memphis, Tennessee, November 17-20, 2009

Arnold P. Boedihardjo, Yao Liang: "Hierarchical Smoothed Round Robin Scheduling in High-Speed Networks," IET Communications Journal, Volume 3, Issue 9, pp. 1557-1568, September 2, 2009

Feng Chen, Chang-Tien Lu, Arnold P. Boedihardjo: "On Locally Linear Classification by Pair-wise Coupling," in Proceedings of the 8th IEEE International Conference on Data Mining (IEEE ICDM), pp. 749-754, Pisa, Italy, December 15-19, 2008

Arnold P. Boedihardjo, Chang-Tien Lu, Feng Chen: "A Framework for Estimating Complex Probability Density Structures in Data Streams," in Proceedings the of 17th ACM Conference on Information and Knowledge Management (ACM CIKM), pp. 619-628, Napa Valley, California, October 26-30, 2008

Chang-Tien Lu, Arnold P. Boedihardjo, David Dai, Feng Chen: "HOMES: Highway Operations and Monitoring and Evaluation System," in Proceedings of the 16th ACM International Conference on Advances in Geographic Information Systems (ACM GIS), pp. 529-530, Irvine, California, November 5-7, 2008

Chang-Tien Lu, Arnold P. Boedihardjo, Shashi Shekhar: "Analysis of Spatial Data with Map Cube: Highway Traffic Data," Geographic Data Mining and Knowledge Discovery, Taylor and Francis, pp. 69-97, 2008

Arnold P. Boedihardjo: "Spatial Stream Mining," Encyclopedia of Geographic Information Science, pp. 1141-1145, 2008

Chang-Tien Lu, Arnold P. Boedihardjo, Jinping Zheng: "Towards an Advanced Spatio-Temporal Visualization System for the Metropolitan Washington D.C.," in Proceedings of the Transportation Research Board International Visualization in Transportation Symposium and Workshop (TRB VIS), Denver, Colorado, October 23-26, 2006

Arnold P. Boedihardjo, Chang-Tien Lu: "AOID: Adaptive Online Incident Detection System," in Proceedings of the 9th IEEE International Conference on Intelligent Transportation System (IEEE ITSC), pp. 858 - 863, Toronto, Canada, September 17-20, 2006

Chang-Tien Lu, Arnold P. Boedihardjo, Jinping Zhen: "AITVS: Advanced Interactive Traffic Visualization System," in Proceedings of the 22nd IEEE International Conference on Data Engineering (IEEE ICDE), pp. 167, Georgia, Atlanta, April 3-8, 2006

Chang-Tien Lu, Arnold P. Boedihardjo, Prajwal Manalwar: "Exploiting Efficient Data Mining Techniques

to Enhance Intrusion Detection System," in Proceedings of the IEEE International Conference on Information Reuse and Integration (IEEE IRI), pp. 512-517, Las Vegas, Nevada, August 15-17, 2005

Posters and Demonstrations

David Dai, Arnold P. Boedihardjo: "HOMES: Highway Operation Monitoring and Evaluation System", at the National Academy of Engineering Grand Challenges Summit Contest, Durham, NC, March 2-3, 2009

Arnold P. Boedihardjo, David Dai, Feng Chen, Chang-Tien Lu: "Enabling Technologies for Providing High-Impact Transportation Network Decision Support for the Washington Metropolitan Area", at the Virginia Council of Graduate Schools 4th Annual Graduate Research Symposium, Richmond, VA, February 10, 2009

Chang-Tien Lu, Arnold P. Boedihardjo: "Advanced Traffic Visualization and Mining System," at the 12th Annual ITSVA (Intelligent Transportation Society of Virginia) Conference and Exhibition, Arlington, VA, June 1-2, 2006

Awards

2nd Place PhD Poster Research Award, Paul E. Torgersen Research Excellence Competition, March 2010

2nd Place Research Poster: "HOMES: Highway Operation Monitoring and Evaluation System", National Academy of Engineering Grand Challenges Summit Contest (Security Category), March 2009

1st Place Research Poster: "Efficient Algorithms for Mining Data Streams", Virginia Tech Northern Capital Region 6th Annual Graduate Research Symposium, Falls Church, 2009

Outstanding Contribution Award: "Enabling Technologies for Providing High-Impact Transportation Network Decision Support for the Washington Metropolitan Area ", Virginia Council of Graduate Schools 4th Annual Graduate Research Forum, February 2009

7.3 Future Work

The future direction of this work will investigate multivariate version of the local region based KDE. Due to the theoretical results established from this research, an efficient approach to local region construction can be realized by applying the SSE criterion and product kernels [80]. Spatial access methods such as the KD-Tree will be considered to efficiently manage the local regions online in multidimensional space. The proposed multi-scale outlier detection framework will also be further investigated to analyze the connection between probability based outliers and other classes of outlier metrics (e.g., distance-based).

Bibliography

[1] "Annual Urban Mobility Report," Texas Transportation Institute [http://mobility.tamu.edu/ums/]2003.

[2] *Freeway Performance Measurement System (PeMS) [http://pems.eecs.berkeley.edu]*.

[3] "Incident Management: Detection, Verification, and Traffic Management," U.S. Department of Transportation Federal Highway Administration1998.

[4] H. Adeli and A. Karim, "Fuzzy-Wavelet RBFNN Model for Freeway Incident Detection," *Journal of Transportation Engineering,* vol. 126, 2000.

[5] C. Aggarwal, "A framework for diagnosing changes in evolving data streams," in proceedings of *2003 ACM SIGMOD International Conference on Management of Data*, San Diego, California, USA, 2003, pp. 575-586.

[6] C. Aggarwal and P. S. Yu, "A survey of synopsis construction in data streams," in *Data Streams: Models and Algorithms*, C. Aggarwal, Ed., ed New York: Springer Science and Business Media, 2007, pp. 169-202.

[7] C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu, "A framework for clustering evolving data streams," *Proceedings of the 2003 Intl. Conf. on Very Large Data Bases (VLDB'03), Berlin, Germany,* 2003.

[8] C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu, "On demand classification of data streams," in proceedings of *10th ACM SIGKDD international conference on Knowledge discovery and data mining*, Seattle, Washington, USA, 2004, pp. 503-508.

[9] A. Arasu, B. Babcock, S. Babu, M. Datar, K. Ito, I. Nishizawa, J. Rosenstein, and J. Widom, "STREAM: the stanford stream data manager (demonstration description)," in proceedings of *2003 ACM SIGMOD international conference on on Management of data*, San Diego, California, USA, 2003, pp. 665-665.

[10] A. Asuncion and D. J. Newman. (2007). *UCI Machine Learning Repository [http://www.ics.uci.edu/~mlearn/MLRepository.html]*.

[11] B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom, "Models and issues in data stream systems," in proceedings of *21st ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, Madison, Wisconsin, USA, 2002, pp. 1-16.

[12] B. Babcock, M. Datar, and R. Motwani, "Load shedding for aggregation queries over data streams," in proceedings of *20th International Conference on Data Engineering*, Boston, Massachusetts, USA, 2004, pp. 350-361.

[13] B. Babcock, M. Datar, and R. Motwani, "Sampling from a moving window over streaming data," in proceedings of *13th Annual ACM-SIAM Symposium on Discrete Algorithms*, San Francisco, California, USA, 2002, pp. 633-634.

[14] B. Babcock, M. Datar, R. Motwani, and L. O'Callaghan, "Maintaining variance and k-medians over data stream windows," in proceedings of *22nd ACM SIGMOD-SIGACT-SIGART symposium on Principles of Database Systems*, San Diego, California, USA, 2003, pp. 234-243.

[15] S. Babu, K. Munagala, J. Widom, and R. Motwani, "Adaptive caching for continuous queries," in

proceedings of *21st International Conference on Data Engineering*, Tokyo, Japan, 2005, pp. 118-129.

[16] S. Babu and J. Widom, "Continuous queries over data streams," *ACM SIGMOD Record,* vol. 30, pp. 109-120, 2001.

[17] Y. Bar-Yam, "Complexity rising, from human beings to human civilization," *UNESCO Encyclopedia of LIfe Support Systems,* 2003.

[18] A. P. Boedihardjo and C.-T. Lu, "AOID: Adaptive On-line Incident Detection System," in proceedings of *International Conference on Intelligent Transportation Systems*, Vancouver, Canada, 2006, pp. 858-863.

[19] A. P. Boedihardjo, C. T. Lu, and F. Chen, "A Framework for Estimating Complex Probability Structures in Data Streams," in proceedings of *17th ACM Conference on Information and Knowledge Management (ACM CIKM)*, Napa Valley, California, 2008, pp. pp. 619-628.

[20] F. Cao, M. Ester, W. Qian, and A. Zhou, "Density-based clustering over an evolving data stream with noise," in proceedings of *6th SIAM International Conference on Data Mining*, Bethesda, Maryland, USA, 2006, pp. 328-339.

[21] C. Chakrabarti, M. Vishwanath, and R. Owens, "Architectures for wavelet transforms: a survey," *The Journal of VLSI Signal Processing,* vol. 14, pp. 171-192, 1996.

[22] M. Charikar, L. O'Callaghan, and R. Panigraphy, "Better streaming algorithms for clustering problems," in proceedings of *25th Annual ACM Symposium on Theory of Computing*, San Diego, California, USA, 2003, pp. 30-39.

[23] D. R. Cox, "Notes on the analysis of mixed frequency distributions," *British Journal of Mathematical and Statistical Psychology,* vol. 19, pp. 39-47, 1966.

[24] Q. Ding, M. Khan, A. Roy, and W. Perrizo, "The P-tree algebra," in proceedings of *2002 ACM symposium on Applied Computing*, Madrid, Spain, 2002, pp. 426-431.

[25] P. Domingos and G. Hulten, "Mining High-Speed Data Streams," in proceedings of *6th ACM Special Interest Group on Knowledge Discovery and Data Mining (SIGKDD)*, Boston, Massachusetts, USA, 2000, pp. 71-80.

[26] J. Duoandikoetxea, *Fourier analysis*: American Mathematical Society, 2001.

[27] M. Ester, H. P. Kriegel, J. Sander, M. Wimmer, and X. Xu, "Incremental clustering for mining in a data warehousing environment," in proceedings of *24th International Conference on Very Large Databases*, New York, USA, 1998.

[28] M. Evanco, "The Impact of Rapid Incident Detection on Freeway Accident Fatalities," U.S. Department of Transportation Federal Highway Administration [http://www.itsdocs.fhwa.dot.gov/jpodocs/repts_te/3FJ01!.pdf]1996.

[29] M. M. Gaber, A. Zaslavsky, and S. Krishnaswamy, "Mining data streams: a review," *ACM SIGMOD Record,* vol. 34, pp. 18-26, 2005.

[30] M. M. Gaber, A. Zaslavsky, and S. Krishnaswamy, "Towards an adaptive approach for mining data streams in resource constrained environments," in proceedings of *6th International Conference on Data Warehousing and Knowledge Discovery*, Zaragoza, Spain, 2004, pp. 189-198.

[31] M. Garofalakis, J. Gehrke, and R. Rastogi, "Querying and mining data streams: you only get one

look (tutorial)," in proceedings of *2002 ACM SIGMOD international conference on Management of data*, Madison, Wisconsin, 2002, pp. 635-635.

[32] P. Gibbons, Y. Matias, and V. Poosala, "Fast incremental maintenance of approximate histograms," *ACM Transactions on Database Systems* vol. 27, pp. 261-298, 2002.

[33] A. Gilbert, Y. Kotidis, S. Muthukrishan, and M. J. Strauss, "How to summarize the universe: dynamic maintenance of quantiles," in proceedings of *28th International Conference of Very Large Data Bases*, Hong Kong, China, 2002, pp. 454-465.

[34] A. Gray and A. Moore, "Rapid evaluation of multiple density models," in proceedings of *9th International Workshop on Artificial Intelligence and Statistics*, Key West, Florida, USA, 2003.

[35] S. Guha, N. Koudas, and K. Shim, "Approximation and streaming algorithms for histogram construction problems," *ACM Transactions on Database Systems,* vol. 31, pp. 396-438, 2006.

[36] S. Guha, A. Meyerson, N. Mishra, and R. Motwani, "Clustering data streams: theory and practice," *IEEE Transactions on Knowledge and Data Engineering,* vol. 15, pp. 515-528, 2003.

[37] S. Guha, N. Mishra, R. Motwani, and L. O'Callaghan, "Clustering data streams," in proceedings of *41st Annual Symposium on Foundations of Computer Science*, Redondo Beach, CA, 2000, p. 359—366.

[38] D. Hand, H. Manilla, and P. Smyth, *Principles of Data Mining*: MIT Press, 2001.

[39] W. Hardle, M. Muller, S. Sperlich, and A. Werwatz, *Nonparametric and semiparametric models*, 1 ed. Germany: Springer-Verlag, 2004.

[40] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning*, 1 ed.: Springer-Verlag, 2001.

[41] C. Heinz, "Density estimation over data streams," Phd, Mathematics, Phillipps-University Marburg, 2007.

[42] C. Heinz and B. Seeger, "Cluster kernels: resource-aware kernel density estimators over streaming data," *IEEE Transactions on Knowledge and Data Engineering,* vol. 20, pp. 880-893, 2008.

[43] C. Heinz and B. Seeger, "Cluster Kernels: Resource-Aware Kernel Density Estimators over Streamining Data," *IEEE Transactions on Knowledge and Data Engineering,* vol. 20, pp. 880-893, 2008.

[44] C. Heinz and B. Seeger, "Exploring data streams with nonparametric estimators," in proceedings of *18th International Conference on Statistical and Scientific Database Management*, Vienna, Austria, 2006, pp. 261-264.

[45] C. Heinz and B. Seeger, "Resource-aware kernel density estimators over streaming data," in proceedings of *15th ACM international conference on Information and knowledge management*, Arlington, VA, USA, 2006, pp. 870-871.

[46] C. Heinz and B. Seeger, "Towards kernel density estimation over streaming data," in proceedings of *13th International Conference on Management of Data*, Delhi, India, 2006, pp. 91-102.

[47] J. Hershberger, N. Shrivastava, and S. Suri, "Cluster Hulls: A Technique for Summarizing Spatial Data Streams," in proceedings of *22nd IEEE International Conference on Data Engineering*, Atlanta, Georgia, USA, 2006, p. 138.

[48] A. Hinneburg and D. Keim, "An efficient approach to clustering in large multimedia databases

with noise," in proceedings of *ACM Knowledge Discovery and Data Mining*, 1998, pp. 58-65.

[49] W. Hoeffding, "Probability inequalities for sums of bounded random variables," *Journal of the American Statistical Association,* vol. 58, pp. 13-30, 1963.

[50] G. Hulten, L. Spencer, and P. Domingos, "Mining time-changing data streams," in proceedings of *7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, California, USA, 2001, pp. 97-106.

[51] Y. Ioannidis, "The history of histograms (abridged)," in proceedings of *29th International Conference on Very Large Databases*, Berlin, Germany, 2003, pp. 19-30.

[52] H. Kargupta, R. Bhargava, K. Liu, M. Powers, P. Blair, S. Bushra, J. Dull, K. Sarkar, M. Klein, M. Vasa, and D. Handy, "VEDAS: A mobile and distributed data stream mining system for real-time vehicle monitoring," in proceedings of *4th SIAM International Conference on Data Mining*, Lake Buena Vista, Florida, 2004, pp. 300-311.

[53] H. Kargupta, B. Park, S. Pittie, L. Liu, D. Kushrai, and K. Sarkar, "MobiMine: Monitoring the stock market from a PDA," *ACM SIGKDD Explorations,* vol. 3, pp. 37-46, 2002.

[54] E. Keogh, X. Xi, L. Wei, and C. A. Ratanamahatana. (2008). *The UCR Time Series Classification/Clustering Database [http://www.cs.ucr.edu/~eamonn/time_series_data]*. Available: http://www.cs.ucr.edu/~eamonn/

[55] S. Kullback and R. A. Leibler, "On information and sufficiency," *Annals of Mathematical Statistics,* vol. 22, pp. 79-86, 1951.

[56] T. Ledl, "Kernel density estimation: theory and application in discriminant analysis," *Austrian Journal of Statistics,* vol. 33, pp. 267-279, 2004.

[57] E. Lehmann, *Theory of point estimation*, 2 ed. New York: Springer, 1998.

[58] C. T. Lu and A. P. Boedihardjo, "AITVS: Advanced Interactive Traffic Visualization System," presented at the 22nd IEEE International Conference on Data Engineering (ICDE 2006), Atlanta, Georgia USA, 2006.

[59] M. W. M. Ferreira, H. Lee, D. Higdon, "A class of multi-scale time series models," ed: Duke University, 2000.

[60] S. Mallat, *A wavelet tour of signal processing*, 3 ed. London: Academic Press, 2008.

[61] T. K. Moon, "The expectation-maximization algorithm," *IEEE Signal Processing,* vol. 13, pp. 47-60, 1996.

[62] J. Natwichai and X. Li, "Knowledge maintenance on data streams with concept drifting," in proceedings of *1st International Symposium on Computation and Information Sciences*, Shanghai, China, 2004, pp. 705-710.

[63] H. J. Nussbaumer, *Fast fourier transform and convolution algorithms*, 2nd ed. New York: Springer-Verlag, 1982.

[64] L. O'Callaghan, N. Mishra, A. Meyerson, S. Guha, and R. Motwani, "Streaming-data algorithms for high-quality clustering," in proceedings of *18th IEEE International Conference on Data Engineering*, San Jose, California, USA, 2002, pp. 685-694.

[65] J. S. Oh, C. Oh, S. G. Ritchie, and M. Chang, "Real-time estimation of accident likelihood for safety enhancement," *Journal of Transportation Engineering,* vol. 131, pp. 358-363, 2005.

[66] I. Ohe, H. Kawashima, M. Kojima, and Y. Kaneko, "A method for automatic detection of traffic

incidents using neural networks," in proceedings of *Vehicle Navigation and Information Systems Conference*, Seattle, Washington, USA, 1995, pp. 231-235.

[67] F. Pan, B. Wang, D. Ren, X. Hu, and W. Perrizo, "Efficient Proximal Support Vector Machine for Spatial Data," in proceedings of *Computer Applications in Industry and Engineering*, Las Vegas, Nevada, USA, 2003, pp. 292-297.

[68] S. Parttikangas, J. Riekki, J. Kaartinen, J. Miettinen, S. Nissila, and J. Roning, "Genie of the net: A new approach for a context-aware health club," in proceedings of *Ubiquitous Data Mining 2001 workshop in conjunction with ECML and PKDD*, Freiburg, Germany, 2001.

[69] E. Parzen, "On estimation of a probability density function and mode," *Annals of Mathematical Statistics,* vol. 33, pp. 1065-1076, 1962.

[70] H. J. Payne and S. C. Tignor, "Freeway Incident Detection Algorithms Based on Decision Tree with States," *Transportation Research Record,* vol. 682, pp. 378-382, 1978.

[71] W. Perrizo, W. Jockheck, A. Perera, D. Ren, W. Wu, and Y. Zhang, "Multimedia data mining using p-trees," in proceedings of *International Workshop on Multimedia Data Mining*, Edmonton, Canada, 2002, pp. 19-29.

[72] B. N. Persaud, F. L. Hall, and L. M. Hall, "Congestion Identification Aspects of the McMaster Incident Detection Algorithm," *Transportation Research Record,* vol. 1287, pp. 167-175, 1990.

[73] K. R. Rao and P. Yip, *Discrete Cosine Transform: Algorithms, Advantages, Applications*: Academic Press Professional, Inc, 1990.

[74] J. Ruoming and G. Agrawal, "Efficient decision tree construction on streaming data," in proceedings of *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Washington, DC, USA, 2003, pp. 571-576.

[75] H. Sagan, *Space-filling curves*: Springer-Verlag, 1994.

[76] S. Sain, "Adaptive kernel density estimation," PhD, Statistics, Rice University, Houston, 1994.

[77] S. R. Sain and D. W. Scott, "On locally adaptive density estimation," *Journal of the American Statistical Association,* vol. 91, pp. 1525-1534, 1996.

[78] H. Samet, "The quadtree and related hierarchical data structures," *ACM Computing Surveys,* vol. 16, pp. 187-260, 1984.

[79] O. Sawaya, "Real-time Incident Traffic Management Methodologies," Ph.D., Civil Engineering, Northwestern University, Evanston, Illinois, 2000.

[80] D. W. Scott, *Multivariate density estimation*. New York: Wiley & Sons, 1992.

[81] R. Shah, S. Krishnaswamy, and M. M. Gaber, "Resource aware very fast k-means for ubiquitous data stream mining," in proceedings of *2nd International Workshop on Knowledge Discovery in Data Streams*, Porto, Portugal, 2005.

[82] B. W. Silverman, *Density estimation for statistics and data analysis*. London: Chapman and Hall, 1986.

[83] A. Skabardonis, P. Varaiya, and K. Petty, "Measuring Recurrent and Non-Recurrent Traffic Congestion," *Transportation Research Record,* vol. 1856, pp. 118-124, 2002.

[84] D. Srinivasan, R. L. Cheu, and Y. P. Poh, "Hybrid Fuzzy Logic-Genetic Algorithm for Automated Detection of Traffic Incidents on Freeways," in proceedings of *IEEE Intelligent Transportation Systems Conference Proceedings*, Oakland, California, USA, 2001, pp. 352-357.

[85] U. Srivastava, S. Babu, and J. Widom, "Monitoring stream properties for continuous query processing," in proceedings of *Workshop on Management and Processing of Data Streams*, San Diego, California, USA, 2003, pp. 1-3.

[86] S. Subramaniam, T. Palpanas, D. Papadopoulos, V. Kalogeraki, and D. Gunopulos, "Online outlier detection in sensor data using non-parametric models," in proceedings of *32nd International Conference on Very Large Databases*, Seoul, Korea, 2006, pp. 187-198.

[87] S. Tanner, M. Alshyeb, E. Criswell, M. Iyer, A. McDowell, M. McEniry, and K. Regner, "EVE: On-board process planning and execution," in proceedings of *Earth Science Technology Conference*, Pasadena, California, 2002.

[88] P. Van Kerm, "Adaptive kernel density estimation," *Stata Journal,* vol. 3, pp. 148-156, 2003.

[89] M. P. Wand and M. C. Jones, *Kernel smoothing*: Chapman and Hall/CRC Press, 1995.

[90] B. Wang, F. Pan, D. Ren, Y. Cui, Q. Ding, and W. Perrizo, "Efficient OLAP operations for spatial data using peano trees," in proceedings of *8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, San Diego, California, USA, 2003, pp. 28-34.

[91] E. J. Wegman and D. J. Marchette, "On some techniques for streaming data: a case study of internet packet headers," *Journal of Computational and Graphical Statistics,* vol. 12, pp. 1-22, 2003.

[92] H. Xu, C. M. Kwan, L. Haynes, and J. D. Pryor, "Real-time adaptive on-line traffic incident detection," in proceedings of *IEEE Symposium on Intelligent Control*, Dearborn, Michigan, USA, 1996, pp. 200-205.

[93] Y. Yao and J. E. Gehrke, "The cougar approach to in-network query processing in sensor networks," *ACM Special Interest Group on Data Management of Data (SIGMOD) Record,* vol. 31, pp. 9-18, 2002.

[94] L. Zhang, Z. Zhu, K. Jeffay, J. S. Marron, and F. Smith, "Multi-resolution anomaly detection for the internet," in proceedings of *1st IEEE Workshop on Automated Network Management*, Phoenix, AZ, 2008.

[95] L. Zhang, Z. Zhu, and J. S. Marron. (2008, MultiResolution Anomaly Detection method for Long Range Depedent Time Series. *Electronic Journal of Statistics 0809(1281v1).*

[96] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: an efficient data clustering method for very large databases," in proceedings of *1996 ACM SIGMOD International Conference on Management of Data*, Montreal, Canada, 1996, pp. 103-114.

[97] T. Zhang, R. Ramakrishnan, and M. Livny, "Fast density estimation using CF-kernel for very large databases," in proceedings of *5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Diego, CA, USA, 1999, pp. 312-316.

[98] A. Zhou, Z. Cai, L. Wei, and W. Qian, "M-Kernel merging: towards density estimation over data streams," in proceedings of *8th International Conference on Database Systems for Advanced Applications*, Kyoto, Japan, 2003, pp. 285-292.

Appendix A. Experimental Settings

This section describes the benchmark dataset, performance measures, and platform employed to conduct the empirical validations of the density estimators.

Experimental Platform

The experiments were performed on a Windows Server 2003 Enterprise OS under .NET 2.0. The hardware platform was a 2.0 GHz Intel Pentium Core 2 Duo with 3 GB of RAM.

Benchmark dataset

The dataset is comprised of both synthetically generated and real-world traces of time series observations. In combination, the dataset reflects a wide range of streaming scenarios which includes distributions with simple structures (e.g., unimodal) to distributions with complex and highly varied features (e.g., multi-scaled and multimodal). The dataset also encompassed varying stationary properties with different periodicity (e.g., power demand load (POWER) and traffic volume (TRAFFIC)). A detailed description of the datasets is shown in the table below:

Name	Type	Description	Size
MIX1	Synthetic	5 time series randomly generated from the following mixture of Normal distributions with randomly selected μ and σ: $F_1 = N(\mu, \sigma^2)$	25K / time series
MIX2	Synthetic	5 time series randomly generated from the following mixture of Normal distributions with randomly selected μ_i and σ_i: $F_2 = \frac{1}{2}\sum_{i=1}^{2} N(\mu_i, \sigma_i^2)$	25K / time series
MIX4	Synthetic	5 time series randomly generated from the following mixture of Normal distributions with randomly selected μ_i and σ_i: $F_4 = \frac{1}{4}\sum_{i=1}^{4} N(\mu_i, \sigma_i^2)$	25K / time series
MIX8	Synthetic	5 time series randomly generated from the following mixture of Normal distributions with randomly selected μ_i and σ_i: $F_8 = \frac{1}{8}\sum_{i=1}^{8} N(\mu_i, \sigma_i^2)$	25K / time series

COMPLEX1	Synthetic	10 time series randomly generated from the following mixture of Normal distributions with P number of modes with regularly spaced centers μ_i, fixed scales σ, and identical mode weights w: $w\sum_{i=1}^{P} N(\mu_i, \sigma^2)$ where $w = 1/P$ for $P = 1..10$	25K / time series
COMPLEX2	Synthetic	10 time series randomly generated from the following mixture of Normal distributions with P number of modes with regularly spaced centers μ_i, varying scales σ_i, and identically mode weights w: $w\sum_{i=1}^{P} N(\mu_i, \sigma_i^2)$ where $w = 1/P$ for $P = 1..10$	25K / time series
COMPLEX3	Synthetic	10 time series randomly generated from the following mixture of Normal distributions with P number of modes with regularly spaced centers μ_i, varying scales σ_i, and varying mode weights w_i: $w_i\sum_{i=1}^{P} N(\mu_i, \sigma_i^2)$ where $\sum_{i=1}^{p} w_i = 1$ for $P = 1..10$	25K / time series
DRIFT1	Synthetic	10 time series randomly generated from the following mixture of Normal distributions with 2 modes with varying centers μ_i, fixed scales σ, and identical mode weights: $\frac{1}{2}\sum_{i=1}^{2} N(\mu_i^j, \sigma^2)$ where $\mu_i < \mu_{i+1}$ and $\lvert\mu_1^r - \mu_2^r\rvert \neq \lvert\mu_1^s - \mu_2^s\rvert$ for all $r \neq s$ and $j = 1..10$	25K / time series
DRIFT2	Synthetic	10 time series randomly generated from the following mixture of Normal distributions with 3 modes with varying centers μ_i, fixed scales σ, and identical mode weights: $\frac{1}{3}\sum_{i=1}^{3} N(\mu_i^j, \sigma^2)$ where $\mu_i < \mu_{i+1}$ and $(\lvert\mu_1^r - \mu_2^r\rvert \neq \lvert\mu_1^s - \mu_2^s\rvert \wedge \lvert\mu_2^r - \mu_3^r\rvert \neq \lvert\mu_2^s - \mu_3^s\rvert)$ for all $r \neq s$ and $j = 1..10$	25K / time series
DRIFT3	Synthetic	10 time series randomly generated from the following mixture of Normal distributions with 4 modes with varying centers μ_i, fixed scales σ, and identical mode weights: $\frac{1}{4}\sum_{i=1}^{4} N(\mu_i^j, \sigma^2)$ where $\mu_i < \mu_{i+1}$ and $(\lvert\mu_1^r - \mu_2^r\rvert \neq \lvert\mu_1^s - \mu_2^s\rvert \wedge \lvert\mu_2^r - \mu_3^r\rvert \neq \lvert\mu_2^s - \mu_3^s\rvert \wedge \lvert\mu_3^r - \mu_4^r\rvert \neq \lvert\mu_3^s - \mu_4^s\rvert)$ for all $r \neq s$ and $j = 1..10$	25K / time series
EEG	Real	EEG of a rat in wake/sleep cycle [54]	25K
POWER	Real	Power demand from a Dutch research facility [54]	25K

ROBOT	Real	Accelerometer measurements of a Sony Aibo Robot playing soccer [54]	24.5K
TRAFFIC	Real	Traffic highway volume readings from a loop detector near a California baseball stadium [2, 10]	25K
INCIDENT	Real	3 time series data from highway volume, speed, and occupancy readings each from a detector station (loop detector aggregator) in Northern Virginia I-66 between January 2004 to July 2005	165K / time series

Table A.1 Dataset descriptions

Performance measures

The table below provides the performance metrics used in the experiments to evaluate the various estimators. Aspects of the estimation quality were measured in terms of both L_1 and L_2 criterions at both local (i.e., point-wise) and global scales. Sample processing throughput was also measured to test the rates for the estimators construct/maintain their internal KDE models. Lastly, the query throughput was measured to determine the rate at which the estimators can generate density estimates.

Name	Type	Description
RDEV	Estimation Quality	Point-wise L_1 estimation discrepancy. Defined as the average relative difference between two estimators: $RDEV(\hat{f}^{(1)}, \hat{f}^{(2)}) = \frac{1}{T}\sum_{i=1}^{T}\left(\frac{ABS\left(\hat{f}^{(1)}(x_i)-\hat{f}^{(2)}(x_i)\right)}{\hat{f}^{(2)}(x_i)}\right)$ where $x_1 .. x_T$ are query points which uniformly divide the entire span of the distribution
IAE	Estimation Quality	Global L_1 estimation discrepancy. Defined as the integrated absolute error spanning the estimation domain and provides the basis of NCI accuracy measure: $IAE(\hat{f}) = \sum_{i=0}^{T} ABS\left(\hat{f}(x_i) - f(x_i)\right)\Delta_i$ where $x_1 .. x_T$ are query points which uniformly divide the entire span of the distribution and Δ_i is the interval width defined starting at point x_i
NCI	Estimation Quality	Global L_1 estimation accuracy. Defined as the normalized complement of IAE (i.e., accuracy): $A(\hat{f}) = 1 - IAE(\hat{f})/E_{max}$ where E_{max} is the theoretical maximum of $IAE(\hat{f})$

RMSE	Estimation Quality	Point-wise L_2 estimation discrepancy. Defined as the root mean squared error spanning the estimation domain: $$RMSE(\hat{f}^{(\rho)}) = \sqrt{\frac{1}{T}\sum_{i=1}^{T}\left(f(x_i) - \hat{f}^{(\rho)}(x_i)\right)^2}$$ where $\hat{f}^{(\rho)}(\cdot)$ is the ρ density estimation technique, $x_1 .. x_T$ are query points which uniformly divide the entire span of the distribution, and $T > 1$
ISE	Estimation Quality	Global L_2 estimation discrepancy. Defined as the integrated squared error spanning the estimation domain: $$ISE(\hat{f}) = \sum_{i=1}^{T}\Delta_i\left(f(x_i) - \hat{f}(x_i)\right)^2$$ where $x_1 .. x_T$ are ordered query points which uniformly divide the entire span of the distribution and Δ_i is the interval width defined starting at point x_i
PR	Throughput	Data sample processing rate. Defined as the rate of data stream samples that can be processed for a given time unit.
PCT	Throughput	Cumulative data sample processing time. Defined as the total amount of time required to insert and process a given set of data points. This measures the efficiency of the estimators in updating its kernel structures to match the current stream.
QR	Throughput	Density query processing rate. Defined as the rate of density queries that can be completed for a given time unit.
QCT	Throughput	Cumulative query processing time. Defined as the total amount of time needed to process the total query set.
TPR	Outlier Detection Quality	True positive rate. Defined as the ratio of the number of the correctly classified outliers to the total number of outliers.
FPR	Outlier Detection Quality	False positive rate. Defined as 1 – (the ratio of correctly classified non-outliers to the total number of non-outliers).

Table A.2 Performance metrics