Change Detection in Streaming Data in the Era of Big Data: Models and Issues

Dang-Hoan Tran
Vietnam Maritime University
484 Lach Tray, Ngo Quyen
Haiphong, Vietnam
dang-hoan.tran@vimaru.edu.vn

Mohamed Medhat Gaber
School of Computing Science and Digital Media, Robert Gordon University
Riverside East Garthdee Road
Aberdeen
AB10 7GJ, UK
m.gaber1@rgu.ac.uk

Kai-Uwe Sattler
Ilmenau University of Technology
PO Box 100565
Ilmenau, Germany
kus@tu-ilmenau.de

ABSTRACT

Big Data is identified by its three *Vs*, namely velocity, volume, and variety. The area of data stream processing has long dealt with the former two *Vs* velocity and volume. Over a decade of intensive research, the community has provided many important research discoveries in the area. The third *V* of Big Data has been the result of social media and the large unstructured data it generates. Streaming techniques have also been proposed recently addressing this emerging need. However, a hidden factor can represent an important fourth *V*, that is variability or change. Our world is changing rapidly, and accounting to variability is a crucial success factor. This paper provides a survey of change detection techniques as applied to streaming data. The review is timely with the rise of Big Data technologies, and the need to have this important aspect highlighted and its techniques categorized and detailed.

1. INTRODUCTION

Today's world is changing very fast. The changes occur in every aspects of life. Therefore, the ability to detect, adapt, and react to the change play an important role in all aspects of life. The physical world is often represented in some model or some information system. The changes in the physical world are reflected in terms of the changes in data or model built from data. Therefore, the nature of data is changing.

The advance of technology results in the data deluge. The data volume is increasing with an estimated rate of 50% per year [39]. Data flood makes traditional methods including traditional distributed framework and parallel models inappropriate for processing, analyzing, storing, and understanding these massive data sets. Data deluge needs a new generation of computing tools that Jim Gray calls the 4^{th} paradigm in scientific computing [25]. Recently, there have been some emerging computing paradigms that meet the requirements of Big Data as follows. Parallel batch processing model only deals with the stationary massive data [17]. However, evolving data continuously arrives with high speed. In fact, online data stream processing is the main approach to dealing with the problem of three characteristics of Big Data including big volume, big velocity, and big variety. Streaming data processing is a model of Big Data processing. Streaming data is temporal data in nature. In addition to the temporal nature, streaming data may include spatial characteristics. For example, geographic information systems can produce spatial-temporal data stream. Streaming data processing and mining have been deploying in real-world systems such as InforSphere Streams (IBM)[1], Rapidminer Streams Plugin [2], StreamBase [3], MOA [4], AnduIN [5]. In order to deal with the high-speed data streams, a hybrid model that combines the advantages of both parallel batch processing model and streaming data processing model is proposed. Some projects for such hybrid model include S4 [6], Storm [7], and Grok [8] .

One of these challenges facing data stream processing and mining is the changing nature of streaming data. Therefore, the ability to identify trends, patterns, and changes in the underlying processes generating data contributes to the success of processing and mining massive high-speed data streams.

A model of continuous distributed monitoring has been recently proposed to deal with streaming data coming from multiple sources. This model has many observers where each observer monitors a single data stream. The goal of continuous distributed monitoring is to perform some tasks that need to aggregate the incoming data from the observers. The continuous distributed monitoring is applied to monitor networks such as sensor networks, social networks, networks of ISP [11].

Change detection is the process of identifying differences in the state of an object or phenomenon by observing it at different times or different locations in space. In the streaming context, change detection is the process of segmenting a data stream into different segments by identifying the points where the stream dynamics change [53]. A change detection method consists of the following tasks: change detection and localization of change. Change detection identifies whether a change occurs, and responds to the presence of such change. Besides change detection, localization of changes determines the location of change. The problem of locating the change has been studied in statistics in the problems of change point detection.

This paper presents the background issues and notation relevant to the problem of change detection in data streams.

2. CHANGE DETECTION IN STREAMING DATA

[1] http://www-01.ibm.com/software/data/infosphere/streams/
[2] http://www-ai.cs.uni-dortmund.de/auto?self=$eit184kc
[3] http://www.streambase.com/
[4] http://moa.cs.waikato.ac.nz/
[5] http://www.tu-ilmenau.de/dbis/research/anduin/
[6] http://incubator.apache.org/s4/
[7] https://github.com/nathanmarz/storm/wiki/Tutorial
[8] https://www.numenta.com/grok_info.html

Streaming computational model is considered one of the widely-used models for processing and analyzing massive data. Streaming data processing helps the decision-making process in real-time. A data stream is defined as follows.

DEFINITION 1. *A data stream is an infinite sequence of elements*

$$S = \{(X_1, T_1), .., (X_j, T_j), ...\} \quad (1)$$

Each element is a pair (X_j, T_j) *where* X_j *is a d-dimensional vector* $X_j = (x_1, x_2, ..., x_d)$ *arriving at the time stamp* T_j*. Time-stamp is defined over discrete domain with a total order. There are two types of time-stamps: explicit time-stamp is generated when data arrive; implicit time-stamp is assigned by some data stream processing system.*

Streaming data includes the fundamental characteristics as follows. First, data arrives continuously. Second, streaming data evolves overtime. Third, streaming data is noisy, corrupted. Forth, timely interfering is important. From the characteristics of streaming data and data stream model, data stream processing and mining pose the following challenges. First, as streaming data arrives rapidly, the techniques of streaming data process and analysis must keep up with the data rate to prevent from the loss of important information as well as avoid data redundancy. Second, as the speed of streaming data is very high, the data volume overcomes the processing capacity of the existing systems. Third, the value of data decreases over time, the recent streaming data is sufficient for many applications. Therefore, one can only capture and process the data as soon as it is generated.

2.1 Change Detection: Definitions and Notation

This section presents concepts and classification of changes and change detection methods. To develop a change detection method, we should understand what a change is.

DEFINITION 2. *Change is defined as the difference in the state of an object or phenomenon over time and/or space [52; 1].*

In the view of system, change is the process of transition from a state of a system to another. In other words, a change can be defined as the difference between an earlier state and a later state. An important distinction between change and difference is that a change refers to a transition in the state of an object or a phenomenon overtime while the difference means the dissimilarity in the characteristics of two objects. A change can reflect the short-term trend or long-term trend. For example, a stock analyst may be interested in the short-term change of the stock price.

Change detection is defined as the process of identifying differences in the state of an object or phenomenon by observing it at different times [54]. In the above definition, a change is detected on the basis of differences of an object at different times without considering the differences of an object in locations in space. In many real world applications, changes can occur both in terms of both time and space. For example, multiple spatial-temporal data streams representing triple (latitude, longitude, time) are created in traffic information systems using GPS [23]. Hence, change detection can be defined as follows.

DEFINITION 3. *Change detection is the process of identifying differences in the state of an object or phenomenon by observing it at different times and/or different locations in space.*

A distinction between concept drift detection and change detection is that concept drift detection focuses on the labeled data while change detection can deal with both labeled and unlabeled data. Change analysis both detects and explains the change. Hido et al. [26] proposed a method for change analysis by using supervised learning.

DEFINITION 4. *Change point detection is identifying time points at which properties of time series data change[32]*

Depending on specific application, change detection can be called in different terms such as burst detection, outlier detection, or anomaly detection. Burst detection a special kind of change detection. Burst is a period on stream with aggregated sum exceeding a threshold [31]. Outlier detection is a special kind of change detection. Anomaly detection can be seen as a special type of change detection in streaming data.

To find a solution to the problem of change detection, we should consider the aspects of change of the system in which we want to detect. As shown in [52], the following aspects of change, which must be considered, include subject of change, type of change, cause of change, effect of change, response of change, temporal issues, and spatial issues. In particular, to design an algorithm for detecting changes in sensor streaming data, the major questions we need to answer include: What is the system in which the changes need to be detected? What are the principles used to model the problem? What is data type? What are the constraints of the problem? What is the physical subject of change? What is the meaning of change to the user? How to respond and react to this change? How to visualize this change?

A change detection method can fall into one of two types: batch change detection and sequential change detection. Given a sequence of N observations $x_1, .., x_N$, where N is invariant, the task of a batch change detection method is deciding whether a change occurs at some point in the sequence by using all N available observations. When the arriving speed of data is too high, batch change detection is suitable. In other words, change detection method using two adjacent windows model will be used. However, the drawback of batch change detection method is that its running time is very large when detecting changes in a large amount of data. In contrast, the sequential change detection problem is based on the observations so far. If no change is detected, the next observation is processed. Whenever a change is detected, the change detector is reset.

Change detection methods can be classified into the following approaches: threshold-based change detection method; state-based change detection method; trend-based change detection method.

A change detection algorithm should meet three main requirements [37]: accuracy, promptness, and online. The algorithm should detect as many as possible actual change points and generate as few as possible false alarms. The algorithm should detect change point as early as possible. The algorithm should be efficient sufficient for a real time environment.

Change detection in data stream allows us to identify the time-evolving trends, and time-evolving patterns. Research issues on mining changes in data streams include modeling and representation of changes, change-adaptive mining method, and interactive exploration of changes [19]. Change detection plays an important role in the field of data stream analysis. Since change in model may convey interesting time-dependent information and knowledge, the change of the data stream can be used for understanding the nature of several applications. Basically, interesting research problems on mining changes in data streams can be classified into three categories: modeling and representation of changes, mining methods, and interactive exploration of changes. Change detection algorithm can be used as a sub-procedure in many other data stream mining algorithms in order to deal with the changing data in data streams [28; 4]. A definition of change detection for streaming data is given as follows

DEFINITION 5. *Change detection is the process of segment-*

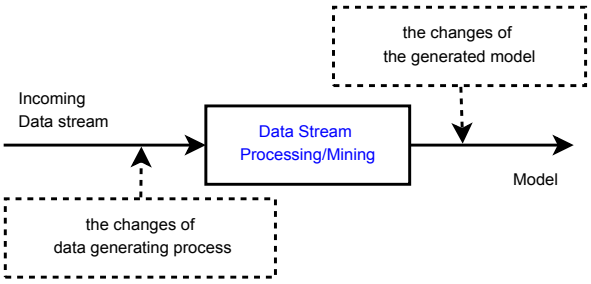


Figure 1: A general diagram for detecting changes in data stream

ing a data stream into different segments by identifying the points where the stream dynamics changes [53].

As data streams evolve overtime in nature, there is growing emphasis on detecting changes not only in the underlying data distribution, but also in the models generated by data stream process and data stream mining. As can be seen in Figure 1, a change can occur in the data stream, or the streaming model. Therefore, there are two types of the problems of change detection: change detection in the data generating process and change detection in the model generated by a data stream processing, or mining. The fundamental issues of detecting changes in data streams includes characterizing and quantifying of changes and detecting changes. A change detection method in streaming data needs a trade-off among space-efficiency, detection performance, and time-efficiency.

2.2 Change Detection Methods in Streaming Data

Over the last 50 years, change detection has been widely studied and applied in both academic research and industry. For example, it has been studied for a long time in the following fields: statistics, signal processing, and control theory. In recent years many change detection methods have been proposed for streaming data. The approaches to detecting changes in data stream can be classified as follows.

- ***Data stream model:*** A data stream can fall into one of the following models: time series model, cash register model, and turnstile model [41]. On the basis of the data stream model, there are change detection algorithms developed for the corresponding data stream model. Krishnamurthy et al presented a sketch-based change detection method for the most general streaming model Turnstile model [35].

- ***Data characteristics:*** Change detection methods can be classified on the basis of the data characteristics of streaming data such as data dimensionality, data label, and data type. A data item coming from the data stream can be univariate or multi-dimensional. It would be great if we could develop a general algorithm able to detect changes in both univariate and multidimensional data streams. Change detection algorithms in streaming multivariate data have been presented [14; 34; 36]. Data streams can be classified into categorial data stream and numerical data stream. We can develop the change detection algorithm for categorial data stream or numerical data stream. In real world applications, each data item in data stream may include multiple attributes of both numerical and categorial data. In such situations, these data streams can be projected by each attribute or group of attributes. Change detection methods can be applied to the corresponding projected data streams afterwards. Data streams are classified into labeled data stream and unlabeled data streams. A labeled data stream is one whose individual example is associated with a given class label, otherwise, it is unlabeled data stream. A change detection algorithm that identifies changes in the labeled data stream is supervised change detection [34; 5], while one detecting changes in the unlabeled data stream is called unsupervised change detection algorithm [7]. The advantage of the supervised approach is that the detection accuracy is high. However, the ground truth data must be generated. Thus a unsupervised change detection approach is preferred to the supervised one in case the ground truth data is unavailable.

- ***Completeness of statistical information:*** On the basis of the completeness of statistical information, a change detection algorithm can fall into one of three following categories. Parametric change detection schemes are based on knowing the full prior information before and after change. For example, in the distributional change detection methods, the data distributions before and after change are known [41; 42]. A recently introduced method to detecting changes in order stock streams is a parametric method in which the distribution of stream of stock orders confide to the Poisson distribution [37]. The advantage of parametric change detection approaches is that they can produce a higher accurate result than semi-parametric and nonparametric methods. However, in many real-time applications, data may not confine to any standard distribution, thus parametric approaches are inapplicable. Semi-parametric methods are based on the assumption that the distribution of observations belongs to some class of distribution function, and parameters of the distribution function change in disorder moments. Recently, Kuncheva [36] has proposed a semi-parametric method using a semi-parametric log-likelihood for testing a change. Nonparametric methods make no distribution assumptions on the data. Nonparametric methods for detecting changes in the underlying data distribution includes Wilcoxon, kernel method, Kullback-Leiber distance, and Kolmogorov-Smirnov test. Nonparametric methods can be classified into two categories: nonparametric methods using window [33]; nonparametric methods without using window [27]. We have paid particular attention to the nonparametric change detection methods using window because in many real-world applications, the distributions of both null hypothesis and alternative hypothesis are unknown in advance. Furthermore, we are only interested in recent data. A common approach to identifying the change is to comparing two samples in order to find out the difference between them, which is called two-sample change detection, or window-based change detection. As data stream is infinite, a sliding window is often used to detect changes. Window based change detection incurs the high delay [37]. Window-based change detection scheme is based on the dissimilarity measure between two distributions or synopses extracted from the reference window and the current window.

- ***Velocity of data change:*** Aggarwal proposes a framework that can deal with the changes in both spatial velocity profile and temporal velocity profile [1; 2]. In this approach, the changes in data density occurring at each location are estimated by estimating velocity density in some user-defined temporal window. An important advantage of this approach is that it visualizes the changes. This visualization of changes helps user understand the changes intuitively.

- ***Speed of response:*** If a change detection method needs to react to the detected changes as fast as possible, the

quickest detection of change should be proposed. Quickest change detection can help a system make a timely alarm. Timely alarm warning is benefit for economical. In some cases, it may save the human life such as in fire-fighting system. Change detection methods using two overlapping windows can quickly react to the changes in streaming data while methods using adjacent windows model may incur the high delay. As change can be abrupt change or gradual change, there exists the abrupt change detection algorithm and gradual change detection algorithm [46; 40].

- ***Decision making methodology:*** Based on the decision making methodology, a change detection method can fall into one of the following categories: rank-based method [33], density-based method [55], information-theoretic method [15]. A change detection problem can be also classified into batch change detection and sequential change detection. Based on detection delay that a change detector suffers from, a change detection methods can fall into one of two following types: real-time change detection, and retrospective change detection. Based on the spatial or temporal characteristics of data, change detection algorithm can fall into one of three kinds: spatial change detection; temporal change detection; or spatio-temporal change detection [6].

- ***Application:*** On the basis of applications that generate data streams, data streams can be classified as into transactional data stream, sensor data stream, network data stream, stock order data stream, astronomy data stream, video data stream, etc. Based on the specific applications, there are the change detection methods for the corresponding applications such as change detection methods for sensor streaming data [56], change detection methods for transactional streaming data [45; 57; 8]. For example, van Leeuwen and Siebes [57] have presented a change detection method for transactional streaming data based on the principle of Minimum Description Length.

- ***Stream processing methodology:*** Based on methodology for processing data stream, a data stream can be classified into online data stream and off-line data stream [38]. In some work, an online data stream is called a live stream while an off-line data stream is called archived data stream [18]. Online data stream needs to be processed online because of its high speed. Such online data streams include streams of stock ticker, streams of network measurements, and streams of sensor data,etc. Off-line stream is a sequence of updates to warehouses or backup devices. The queries over the off-line streams can be processed off-line. However, as it is insufficient time to process off-line streams, techniques for summarizing data are necessary. In off-line change detection method, the entire data set is available for the analysis process to detect the change. The online method detects the change incrementally based on the recently incoming data item. An important distinction between off-line method and online one is that the online method is constrained by the detection and reaction time due to the requirement of real-time applications while the off-line is free from the detection time, and reaction time. Methods for detecting changes can be useful for streaming data warehouses where both live streams of data and archived data streams are available [24; 29]. In this work, we focus on developing the methods for detecting changes in online data streams, in particular, sensor data streams.

The first work on model-based change detection proposed by [21; 22] is FOCUS. The central idea behind FOCUS is that the models can be divided into structural and measurement components. To detect deviation between two models, they compare specific parts of these corresponding models. The models obtained by data mining algorithms includes frequent item sets, decision trees, and clusters. The change in model may convey interesting information or knowledge of an event or phenomenon. Model change is defined in terms of the difference between two set of parameters of two models and the quantitative characteristics of two models. As such, model change detection is finding the difference between two set of parameters of two models and the quantitative characteristics of these two models. We should distinguish between detection of changes in data distribution by using models and detection of changes in model built from streaming data. While model change detection aims to identify the difference between two models, change detection in the underlying data distribution by using models is inferring the changes in two data sets from the difference between two models constructed from two data sets. The changes in the underlying data distribution can induce the corresponding changes in the model produced from the data generating process.

As models can be generated by statistics method or data mining methods, change detection in models can be classified into data mining model and statistical model. Two kinds of models we are interested in detecting changes are predictive model and explanatory model. Predictive model is used to predict the changes in the future. Detecting changes in the pattern can be beneficial for many applications. In explanatory model, a change that occurred is both detected and explained. There are some approaches to change detection: one-model approach, two-model approach, or multiple-model approach.

A model-based change detection algorithm consists of two phases as follows: model construction and change detection. First, a model is built by using some stream mining method such as decision tree, clustering, frequent pattern. Second, a difference measure between two models is computed based the characteristics of the model, this step is also called the quantification of model difference. Therefore, one fundamental issue here is to quantify the changes between two models and to determine criteria for making decision whether and when a change in the model occurs. Recently, some change detection methods in streaming data by clustering have been proposed [10; 3]. Based on the data stream mining model, we may have the corresponding problems of detecting changes in model as follows. Ikonomovska et al. [30] have presented an algorithm for learning regression trees from streaming data in the presence of concept drifts. Their change detection method is based on sequential statistical tests that monitoring the changes of the local error, at each node of tree, and inform the learning process of the local changes.

Detecting changes of stream cluster model has been received increasing attention. Zhou et al. [59] have presented a method for tracking the evolution of clusters over sliding windows by using temporal cluster features and the exponential histogram, which called exponential histogram of cluster features. Chen and Liu [9] have presented a framework for detecting the changes in clustering structures constructed from categorial data streams by using hierarchial entropy trees to capture the entropy characteristics of clusters, and then detecting changes in clustering structures based on these entropy characteristics.

Based on the data stream mining model, we may have the corresponding problems of detecting changes in model as follows [14]. Recently Ng and Dash [44] have introduced an algorithm for mining frequent patterns from evolving data streams. Their algorithm is capable of updating the frequent patterns based on the algorithms for detecting changes in the underlying data distributions. Two windows are used for change detection: the reference window and the current window. At the initial stage, the

reference is initialized with the first batch of transactions from data stream. The current window moves on the data stream and captures the next batch of transactions. Two frequent item sets are constructed from two corresponding windows by using the Apriori algorithm. A statistical test is performed on two absolute support values that are computed by the Apriori from the reference window and current window. Based on the statistical test, the deviation can be significant or insignificant. If the deviation is significant then a change in the data stream is reported. Chang and Lee [8] have presented a method for monitoring the recent change of frequent item sets from data stream by using sliding window.

2.3 Design Methodology

There are two design methodologies for developing the change detection algorithms in streaming data. The first methodology is to adapt the existing change detection methods for streaming data. However, many traditional change detection methods cannot be extended for streaming data because of the high computational complexity such as some kernel-based change detection methods, and density-based change detection methods. The second methodology is to develop new change detection methods for streaming data.

There are two common approaches to the problem of change detection in streaming data distributions: distance-based change detectors and predictive model-based change detectors. In the former, two windows are used to extract two data segments from the data stream. The change is quantified by using some dissimilarity measure. If the dissimilarity measure is greater than a given threshold then a change is detected. Similar to distance-based change detectors, two windows are used for detecting changes. Instead of comparing the dissimilarity measure between two windows with a given threshold, a change is detected by using the prediction error of the model built from the current window and the predictive model constructed from the reference window.

3. DISTRIBUTED CHANGE DETECTION IN STREAMING DATA

Knowledge discovery from massive amount of streaming data can be achieved only when we could develop the change detection frameworks that monitor streaming data created by multiple sources such as sensor networks, WWW [13]. The objectives of designing a distributed change detection scheme are maximizing the lifetime of the network, maximizing the detection capability, and minimizing the communication cost [58].

There are two approaches to the problem of change detection in streaming data that is created from multiple sources. In the centralized approach: all remote sites send raw data to the coordinator. The coordinator aggregates all the raw streaming data that is received from the remote sites. Detection of changes is performed on the aggregated streaming data. In most cases, communication consumes the largest amount of energy. The lifetime of sensors therefore drastically reduces when they communicate raw measurements to a centralized server for analysis. Centralized approaches suffer from the following problems: communication constraint, power consumption, robustness, and privacy. Distributed detection of changes in streaming data addresses the challenges that come from the problem of change detection, data stream processing, and the problem of distributed computing. The challenges coming from the distributed computing environment are as follows

- Distributed change detection in streaming data is a problem of distributed computing in nature. Therefore, a distributed framework for detecting changes should meet the properties of distributed computing such scalability, and fault tolerance. The scalability refers to the ability to extend the size of the network without significantly reducing the performance of the framework. As faults may occur due to the transmission error and the effects of noisy channels between local sensors and fusion center, a distributed change detection method should be able to tolerate these faults in order to assure the function of the system.

- Distributed change detection using the local approach is directly relevant to the problem of multiple hypotheses testing and data fusion because each local change detector needs to perform a hypothesis test to determine whether a change occurs. Therefore, besides considering the detection performance of local change detection algorithms including probability of detection and probability of false alarm at the node level, the detection performance of a distributed change detection method at the fusion center must be taken into account.

Distributed detection and data fusion have been widely studied for many decades. However, only recently, distributed detection in streaming data has received attention.

3.1 Distributed Detection: One-shot versus Continuous

Distributed detection of changes can be classified into two types of models as follows.

- One-shot distributed detection of changes: Figure 2 shows two models of one-shot distributed change detection. One-shot change detection method means a change detector detects and reacts to the detected change once a change is detected. One-shot distributed change detection have received great deal of attention for a long time. One-shot distributed change detection include two models: distributed detection with decision with decision fusion as shown in Figure 2(a); distributed detection without decision fusion as illustrated in Figure 2(b). What are the differences between one-shot detection and continuous detection.

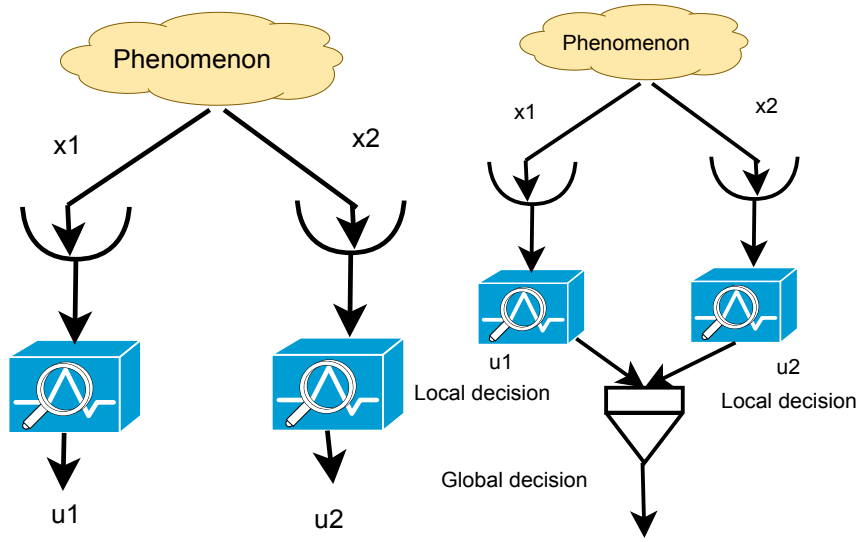
- Continuous distributed detection of changes: In this chapter, we propose two continuous distributed detection models as shown in Figure 3. An important distinction between continuous distributed detection of changes and one-shot distributed detection of changes is that the inputs to the one-shot distributed change detection are batches of data while the inputs to the continuous distributed detection of changes are the data streams in which data items continuously arrive.

Distributed detection model without fusion is a truly distributed detection model in which The decision-making process occurs at each sensor.

3.2 Locality in Distributed Computing

As one of the properties of distributed computational systems is locality [43], a distributed algorithm for detecting changes in streaming data should meet the locality. A local algorithm is defined as one whose resource consumption is independent of the system size. The scalability of distributed stream mining algorithms can be achieved by using the local change detection algorithms

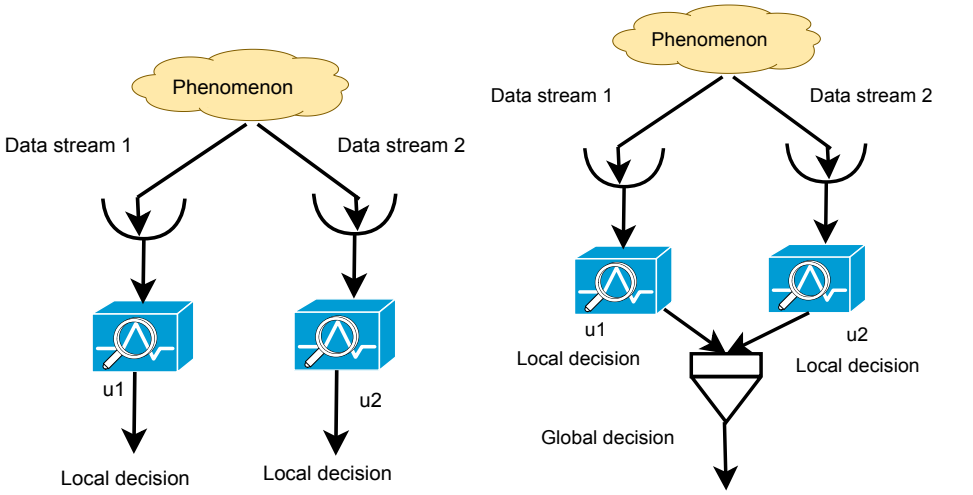
Local algorithms can fall into one of two categories [16]:Exact local algorithms are defined as ones that produce the same results as a centralized algorithm; Approximate local algorithms are algorithms that produce approximations of the results that centralized algorithms would produce. Two attractive properties of local algorithms are scalability and fault tolerance. A distributed



(a) Distributed detection without decision fusion

(b) Distributed detection with decision fusion

Figure 2: One-shot distributed change detection models



(a) Distributed continuous detection without decision fusion

(b) Distributed continuous detection with decision fusion

Figure 3: Continuous distributed change detection models

framework for mining streaming data should be robust to network partitions, and node failures.

The advantage of local approaches is the ability to preserve privacy [20]. A drawback of the local approach to the problem of distributed change detection is the synchronization problem. For example, the local change approach can meet the principle of localized algorithms in wireless sensor networks in which data processing is performed at node-level as much as possible in order to reduce the amount of information to be sent in the network.

3.3 Distributed Detection of Changes in Streaming Data

Over the last decades, the problem of decentralized detection has received much attention. There are two directions of research on decentralized detection. The first approach focuses on aggregating measurements from multiple sensors to test a single hypothesis. The second focuses on dealing with multiple dependent testing/estimation tasks from multiple sensors [51]. Distributed change detection usually involves a set of sensors that receive observations from the environment and then transmit those observations back to fusion center in order to reach the final consensus of detection. Decentralized detection and data fusion are therefore two closely related tasks that arise in the context of sensor networks [48; 47]. Two traditional approaches to the decentralized change detection are data fusion, and decision fusion. In data fusion, each node detects change and sends quantized version of its observation to a fusion center responsible for making decision on the detected changes, and further relaying information. In contrast, in decision fusion, each node performs local change detection by using some local change algorithm and updates its decision based on the received information and broadcasts again its new decision. This process repeats until consensus among the nodes are reached. Compared to data fusion, decision fusion can reduce the communication cost because sensors need only to transmit the local decisions represented by small data structures. Although there is great deal of work on distributed detection and data fusion, most of work focuses on the one-time change detection solutions. One-time query is defined as a query that needs to proceed data once in order to provide the answer [12]. Likewise, one-time change detection method is a change detection that requires to proceed data once in response to the change occurred. In real-world applications, we need the approaches capable of continuously monitoring the changes of the events occurring in the environment. Recently, work on continuous detection and monitoring of changes has been started receiving attention such as [49; 13; 50]. Das et al. [13] have presented a scalable distributed framework for detecting changes in astronomy data streams using local, asynchronous eigen monitoring algorithms. Palpanas et al. [49] proposed a distributed framework for outlier detection in real-time data streams. In their framework, each sensor estimates and maintains a model for its underlying distribution by using kernel density estimators. However, they did not show how to reach the global detection decision.

4. CONCLUDING REMARKS

We argued in this paper that variability, or simply change, is crucial in a world full of affecting factors that alter the behavior of the data, and consequently the underlying model. The ability to detect such changes in centralized as well as distributed system plays an important role in identifying validity of data models.

The paper presented the state-of-the-art in this area of paramount importance. Techniques, in some cases, are tightly coupled with application domains. However, most of the techniques reviewed in this paper are generic and could be adapted to different domains of applications.

With Big Data technologies reaching a mature stage, the future work in change detection is expected to exploit such scalable data processing tools in efficiently detect, localize and classify occurring changes. For example, distributed change detection models can make use of the *MapReduce* framework to accelerate their respective processes.

5. REFERENCES

[1] C. Aggarwal. A framework for diagnosing changes in evolving data streams. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, pages 575–586. ACM New York, NY, USA, 2003.

[2] C. Aggarwal. On change diagnosis in evolving data streams. *IEEE Transactions on Knowledge and Data Engineering*, pages 587–600, 2005.

[3] C. Aggarwal. A segment-based framework for modeling and mining data streams. *Knowledge and information systems*, 30(1):1–29, 2012.

[4] C. Aggarwal and P. Yu. A survey of synopsis construction in data streams. *Data streams: models and algorithms*, page 169, 2007.

[5] A. Bondu and M. Boullé. A supervised approach for change detection in data streams. In *Neural Networks (IJCNN), The 2011 International Joint Conference on*, pages 519–526. IEEE, 2011.

[6] S. Boriah, V. Kumar, M. Steinbach, C. Potter, and S. Klooster. Land cover change detection: a case study. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 857–865. ACM, 2008.

[7] G. Cabanes and Y. Bennani. Change detection in data streams through unsupervised learning. In *Neural Networks (IJCNN), The 2012 International Joint Conference on*, pages 1–6. IEEE, 2012.

[8] J. Chang and W. Lee. estwin: adaptively monitoring the recent change of frequent itemsets over online data streams. In *Proceedings of the twelfth international conference on Information and knowledge management*, pages 536–539. ACM, 2003.

[9] K. Chen and L. Liu. HE-Tree: a framework for detecting changes in clustering structure for categorical data streams. *The VLDB Journal*, pages 1–20.

[10] T. CHEN, C. YUAN, A. SHEIKH, and C. NEUBAUER. Segment-based change detection method in multivariate data stream, Apr. 9 2009. WO Patent WO/2009/045,312.

[11] G. Cormode. The continuous distributed monitoring model. *SIGMOD Record*, 42(1):5, 2013.

[12] G. Cormode and M. Garofalakis. Efficient strategies for continuous distributed tracking tasks. *IEEE Data Engineering Bulletin*, 28(1):33–39, 2005.

[13] K. Das, K. Bhaduri, S. Arora, W. Griffin, K. Borne, C. Giannella, and H. Kargupta. Scalable Distributed Change Detection from Astronomy Data Streams using Local, Asynchronous Eigen Monitoring Algorithms. In *SIAM International Conference on Data Mining, Nevada*, 2009.

[14] T. Dasu, S. Krishnan, D. Lin, S. Venkatasubramanian, and K. Yi. Change (Detection) You Can Believe in: Finding Distributional Shifts in Data Streams. In *Proceedings of the 8th International Symposium on Intelligent Data Analysis: Advances in Intelligent Data Analysis VIII*, page 34. Springer, 2009.

[15] T. Dasu, S. Krishnan, S. Venkatasubramanian, and K. Yi. An information-theoretic approach to detecting changes in multi-dimensional data streams. In *38th Symposium on the Interface of Statistics, Computing Science, and Applications*. Citeseer, 2005.

[16] S. Datta, K. Bhaduri, C. Giannella, R. Wolff, and H. Kargupta. Distributed data mining in peer-to-peer networks. *IEEE Internet Computing*, pages 18–26, 2006.

[17] J. Dean and S. Ghemawat. Mapreduce: Simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008.

[18] N. Dindar, P. M. Fischer, M. Soner, and N. Tatbul. Efficiently correlating complex events over live and archived data streams. In *ACM DEBS Conference*, 2011.

[19] G. Dong, J. Han, L. Lakshmanan, J. Pei, H. Wang, and P. Yu. Online mining of changes from data streams: Research problems and preliminary results. Citeseer.

[20] A. R. Ganguly, J. Gama, O. A. Omitaomu, M. M. Gaber, and R. R. Vatsavai. *Knowledge discovery from sensor data*, volume 7. CRC, 2008.

[21] V. Ganti, J. Gehrke, and R. Ramakrishnan. A framework for measuring changes in data characteristics. In *Proceedings of the eighteenth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 126–137. ACM, 1999.

[22] V. Ganti, J. Gehrke, R. Ramakrishnan, and W. Loh. A framework for measuring differences in data characteristics. *Journal of Computer and System Sciences*, 64(3):542–578, 2002.

[23] S. Geisler, C. Quix, and S. Schiffer. A data stream-based evaluation framework for traffic information systems. In *Proceedings of the ACM SIGSPATIAL International Workshop on GeoStreaming*, pages 11–18. ACM, 2010.

[24] L. Golab, T. Johnson, J. S. Seidel, and V. Shkapenyuk. Stream warehousing with datadepot. In *Proceedings of the 35th SIGMOD international conference on Management of data*, pages 847–854. ACM, 2009.

[25] A. J. Hey, S. Tansley, and K. M. Tolle. *The fourth paradigm: data-intensive scientific discovery*. Microsoft Research Redmond, WA, 2009.

[26] S. Hido, T. Idé, H. Kashima, H. Kubo, and H. Matsuzawa. Unsupervised change analysis using supervised learning. In *Proceedings of the 12th Pacific-Asia conference on Advances in knowledge discovery and data mining*, pages 148–159. Springer-Verlag, 2008.

[27] S. Ho and H. Wechsler. Detecting changes in unlabeled data streams using martingale. In *Proceedings of the 20th international joint conference on Artifical intelligence*, pages 1912–1917. Morgan Kaufmann Publishers Inc., 2007.

[28] W. Huang, E. Omiecinski, and L. Mark. Evolution in Data Streams. 2003.

[29] W. Huang, E. Omiecinski, L. Mark, and M. Nguyen. History guided low-cost change detection in streams. *Data Warehousing and Knowledge Discovery*, pages 75–86, 2009.

[30] E. Ikonomovska, J. Gama, R. Sebastião, and D. Gjorgjevik. Regression trees from data streams with drift detection. In *Discovery Science*, pages 121–135. Springer, 2009.

[31] M. Karnstedt, D. Klan, C. Pölitz, K.-U. Sattler, and C. Franke. Adaptive burst detection in a stream engine. In *Proceedings of the 2009 ACM symposium on Applied Computing*, pages 1511–1515. ACM, 2009.

[32] Y. Kawahara and M. Sugiyama. Change-point detection in time-series data by direct density-ratio estimation. In *Proceedings of 2009 SIAM International Conference on Data Mining (SDM2009)*, pages 389–400, 2009.

[33] D. Kifer, S. Ben-David, and J. Gehrke. Detecting change in data streams. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, page 191. VLDB Endowment, 2004.

[34] A. Kim, C. Marzban, D. Percival, and W. Stuetzle. Using labeled data to evaluate change detectors in a multivariate streaming environment. *Signal Processing*, 89(12):2529–2536, 2009.

[35] B. Krishnamurthy, S. Sen, Y. Zhang, and Y. Chen. Sketch-based change detection: Methods, evaluation, and applications. In *Proceedings of the 3rd ACM SIGCOMM Conference on Internet Measurement*, pages 234–247. ACM New York, NY, USA, 2003.

[36] L. Kuncheva. Change detection in streaming multivariate data using likelihood detectors. *Knowledge and Data Engineering, IEEE Transactions on*, (99):1–1, 2011.

[37] X. Liu, X. Wu, H. Wang, R. Zhang, J. Bailey, and K. Ramamohanarao. Mining distribution change in stock order streams. *Prof. of ICDE*, pages 105–108, 2010.

[38] G. Manku and R. Motwani. Approximate frequency counts over data streams. In *Proceedings of the 28th international conference on Very Large Data Bases*, pages 346–357. VLDB Endowment, 2002.

[39] J. Manyika, M. Chui, B. Brown, J. Bughin, R. Dobbs, C. Roxburgh, and A. Byers. Big data: The next frontier for innovation, competition and productivity. *McKinsey Global Institute, May*, 2011.

[40] A. Maslov, M. Pechenizkiy, T. Kärkkäinen, and M. Tähtinen. Quantile index for gradual and abrupt change detection from cfb boiler sensor data in online settings. In *Proceedings of the Sixth International Workshop on Knowledge Discovery from Sensor Data*, pages 25–33. ACM, 2012.

[41] S. Muthukrishnan. *Data streams: Algorithms and applications*. Now Publishers Inc, 2005.

[42] S. Muthukrishnan, E. van den Berg, and Y. Wu. Sequential change detection on data streams. *ICDM Workshops*, 2007.

[43] M. Naor and L. Stockmeyer. What can be computed locally? pages 184–193, 1993.

[44] W. Ng and M. Dash. A change detector for mining frequent patterns over evolving data streams. In *Systems, Man and Cybernetics, 2008. SMC 2008. IEEE International Conference on*, pages 2407–2412. IEEE, 2008.

[45] W. Ng and M. Dash. A test paradigm for detecting changes in transactional data streams. In *Database Systems for Advanced Applications*, pages 204–219. Springer, 2008.

[46] D. Nikovski and A. Jain. Fast adaptive algorithms for abrupt change detection. *Machine learning*, 79(3):283–306, 2010.

[47] R. Niu and P. K. Varshney. Performance analysis of distributed detection in a random sensor field. *Signal Processing, IEEE Transactions on*, 56(1):339–349, 2008.

[48] R. Niu, P. K. Varshney, and Q. Cheng. Distributed detection in a large wireless sensor network. *Information Fusion*, 7(4):380–394, 2006.

[49] T. Palpanas, D. Papadopoulos, V. Kalogeraki, and D. Gunopulos. Distributed deviation detection in sensor networks. *ACM SIGMOD Record*, 32(4):77–82, 2003.

[50] D.-S. Pham, S. Venkatesh, M. Lazarescu, and S. Budhaditya. Anomaly detection in large-scale data stream networks. *Data Mining and Knowledge Discovery*, 28(1):145–189, 2014.

[51] R. Rajagopal, X. Nguyen, S. C. Ergen, and P. Varaiya. Distributed online simultaneous fault detection for multiple sensors. In *Information Processing in Sensor Networks, 2008. IPSN'08. International Conference on*, pages 133–144. IEEE, 2008.

[52] J. Roddick, L. Al-Jadir, L. Bertossi, M. Dumas, H. Gregersen, K. Hornsby, J. Lufter, F. Mandreoli, T. Mannisto, E. Mayol, et al. Evolution and change in data management:issues and directions. *ACM Sigmod Record*, 29(1):21–25, 2000.

[53] G. Ross, D. Tasoulis, and N. Adams. Online annotation and prediction for regime switching data streams. In *Proceedings of the 2009 ACM symposium on Applied Computing*, pages 1501–1505. ACM, 2009.

[54] A. Singh. Review Article Digital change detection techniques using remotely-sensed data. *International Journal of Remote Sensing*, 10(6):989–1003, 1989.

[55] X. Song, M. Wu, C. Jermaine, and S. Ranka. Statistical change detection for multi-dimensional data. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 667–676. ACM, 2007.

[56] D.-H. Tran and K.-U. Sattler. On detection of changes in sensor data streams. In *Proceedings of the 9th International Conference on Advances in Mobile Computing and Multimedia*, pages 50–57. ACM, 2011.

[57] M. van Leeuwen and A. Siebes. Streamkrimp: Detecting change in data streams. *Machine Learning and Knowledge Discovery in Databases*, pages 672–687, 2008.

[58] V. Veeravalli and P. Varshney. Distributed inference in wireless sensor networks. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 370(1958):100–117, 2012.

[59] A. Zhou, F. Cao, W. Qian, and C. Jin. Tracking clusters in evolving data streams over sliding windows. *Knowledge and Information Systems*, 15(2):181–214, 2008.