Building Adaptive Data Mining Models on Streaming Data in Real-Time

Frederic Stahl[1,2] and Atta Badii[2]

[1] German Research Center for Artificial Intelligence GmbH (DFKI), Marine Perception, Marie-Curie-Straße 1 26129 Oldenburg, Germany
Frederic_Theodor.Stahl@dfki.de
[2] Department of Computer Science, University of Reading
Whiteknights. PO Box 225, Reading, RG6 6AY, UK,
Atta.Badii@reading.ac.uk, F.T.Stahl@reading.ac.uk

Abstract. This article highlights some key concepts and emergent techniques in DSM as presented in the authors' recent publications and also outlined in a talk given at the UK Symposium on Knowledge Discovery from Data in London on May 24th, 2019. This talk discussed the challenges, opportunities and innovative solutions in Data Stream Mining. The Idea for the talk stemmed from advances in hard and software over two decades enabling capturing of data near real-time. Because of this the research field of Data Stream Mining has been growing in order to tackle real-time analytics of Bit Data as it is being generated. There is a need to build and update models in real-time as new data becomes available in order to maintain model accuracy over time. Applications are for example telecommunications data, telemetric data from industry plants, cyber security, micro-blogging data, etc.

Keywords: Data Stream Mining, Concept Drift, Adaptive Data Mining Models

1 Introduction

Data is generated at an unprecedented rate and scale; this is often referred to as the Velocity and Volume of Big Data. The domo website [5] regularly publishes an infographic that sets out examples of how much data certain internet applications produce every minute of the day. For instance, the infographic states for the year 2017 that on average, every minute, around 176000 Skype calls were made, 4.3 Million YouTube videos uploaded and 3.9 Million google searches performed. IBM estimated that worldwide, with the passing of every day 2.5 Quintillion more data were generated. It is difficult to imagine how much 2.5 Quintillion bytes can store [6]. To visualise this consider storing the data on DVD or Blue-ray discs. Even although these media are hardly used nowadays, we still have a perception of how much data could fit on a disc due to its standard storage capacity. A DVD can store 4.7 GB of data, a Blue-ray disc 25GB of data. It would take a 100 Million Blue-ray discs to store 2.5 Quintillion of data and at 1.22 mm thickness per disc this would stack up to about 100km -a distance of roughly from London to the Isle of Wight. On DVD discs this would take about 530 Million discs which at 1.22 mm thickness would stack to about 630km high - a distance from London to Bremen (Germany). Even although this is an estimate and its reliability could be challenged, it can nevertheless be agreed that even if a small fraction of the 2.5 Quintilian data is generated every day, it is still a very large amount of data. In order to make sense of this real-time continuous stream of data, continuously running analytics methods are required. This article is structured as follows: Section 2 defines the concept of Data Streams and introduces challenges and barriers in analysing such real-time streams. Section 3 highlights some general approaches and algorithms for building models for data stream analytics. This is followed in Section 4 by some recent and ongoing research to overcome the barriers; the concluding remarks are provided in Section 5.

2 Real-time Streaming Data Analytics Challenges and Barriers

Advances in data acquisition hardware and the emergence of applications that process continuous flows of data have led to the (Big Data) stream phenomenon. A data stream is a rapid flow of data which demands analytics such that is challenging for the state-of-the-art processing techniques and communication infrastructure. There are some fundamental differences between static and streaming data and these are: (i) static data is typically historic information, whereas streaming data is often a live real-time feed of data; (ii) static data is randomly accessible, whereas streaming data can be accessed only sequentially, one data instance at a time, (iii) static data is located in secondary storage, whereas streaming data needs to be processed in memory to increase efficiency and enable real-time analytics; iv) for static data the time to build analytics models is often not critical, whereas in streaming environments a low processing latency is often essential to enable real-time applications finally (v) when using static data for model building, one can assume the data is already pre-processed, or at least sufficient time is available to clean the data, whereas in a streaming environment one must assume the data includes inaccurate and raw data elements that are yet to be checked prior to model building. This never-ending stream of raw data is often referred to as the Data Tsunami.

2.1 concept Drift

An important challenge in data streams analytics is concept drift, where the pattern encoded in the stream changes over time. Concept drift exists in real life problems, such as seasonal weather changes, stock market downturns and rallies etc. Concept drifts are typically unforeseen and unpredictable. There are different kinds of concept drifts, i.e. gradual drift, where previous and new patterns encoded in the stream overlay for a short period of time, sudden drift, where the pattern in the stream shifts abruptly, the new patterns appears instantly and the old pattern disappears at the same time. Then there are evolving or incremental streams, where the pattern does not stay stable and changes constantly, and, re-occurring drifts, where previously seen and discontinued patterns re-occur. Of course, there are also combinations of these types of streams. A data mining model should always reflect the current concept and thus needs to adapt quickly.

2.2 Challenges and Barriers

Table 1 summarises the challenges in mining real-time data streams. Each challenge is opposed by a barrier which needs to be overcome to address the challenge fully. One of these challenges is concept drift which is the most fundamental challenge in analysing data streams as otherwise data streams could be treated as a batch problem and models could be simply built from a sample of the streamed data, as there would be no need for adaptation in real-time. However, the remaining challenges are also important. Next, Section 3 will summarise some general approaches to analysis of data streams, in particular with respect to challenge 2 (concept drift).

Table 1: Challenges and Barriers in analysing data streams.

No.	Challenges	Barriers
1	Data generated at a fast rate (Velocity), at potentially large and unknown quantities (Volume)	Limited scalable (parallel) real-time data stream mining algorithms available
2	Concept Drifts (Changes of pattern encoded in in the data over time)	Different and changing types of concept drift
3	Real-time data model building workflows optimisation based on streaming data	Lack of customizable pre-processing techniques
4	Multi-modality of data sources (Text, Video/images, (un)structured)	Different time stamps but co-occurring data items
5	Class label sparsity: need to adapt the associated predictive models	Supervised algorithms not applicable in many cases
6	High throughput real-time processing	Limited parallel real-time architectures

3 Building Data Mining Models from Data Streams

A popular technique for data stream mining, which is in some form often used within adaptive Data Stream Mining algorithms or in combination with concept drift detection methods to enable the execution of existing batch Data Mining algorithms on streaming data, is windowing [8,12]. A frequently used technique here, to name an example, is the ADaptive sliding WINdow method (ADWIN) [10]. ADWIN changes the window size based on observed changes: if there are changes then the window shrinks, if there are no changes it increases. Concept drift detection methods are typically used in combination with batch data mining algorithms and sliding window approaches. For example, the sequential analysis technique CUmulative SUM [20] detects a drift when the mean of incoming data deviates significantly. The Exponentially Weighted Moving Average (EWMA) uses charts to monitor the mean of misclassification rates of a classifier, and gives less weight to older data instances and greater weight to more recent data instances. In general, different methods are suitable for detection and adaptation to different kinds of concept drift, e.g. DDM [14] is suitable for sudden drifts and EDDM for gradual drifts [9]. Challenges here are not to mistake outliers and the noise of uncleansed raw data for concept drift.

The following is a brief overview of some adaptive predictive and descriptive data stream mining algorithms that naturally incorporate windowing/data buffering techniques and adaptation to concept drift. All these algorithms have one requirement in common, which is that they must only require a single pass through the training data and have a small memory footprint.

3.1 Adaptive Predictive Algorithms

The development of adaptive classification techniques has received considerable attention in the recent years. These range from adaptive tree-based methods, i.e. the popular Hoeffding Tree [16] algorithm, to rule-based methods such as VFDR [13] and G-eRules [19] to less expressive (but often very fast) methods such as MC-NN [21]. A drawback of most data stream classification methods arises from the lack for a "cold-start" capability which means that they need ground truth information. This is often unrealistic as most applications will not be able to provide this fast feedback about the correctness of the classification and thus these supervised methods are often not suitable for real-life applications (see Barrier 5 in Table 1). However, some applications can provide indirect class information, e.g. for the classification of twitter posts into topics categories, one could use hashtags instead of the actual categories, to verify and adapt the classification model.

3.2 Descriptive Adaptive Cluster Analysis Algorithms

Cluster analysis is the method of grouping data with similar characteristics, with the aim of having a high degree of similarity within a cluster, but a high degree of dissimilarity between clusters (optimal sensitivity and selectivity based on inter-intra class distances). Several such algorithms exist for streaming data environments. Most of these are built on the idea of Micro-Cluster structures; these are statistical summaries and typically as many as computationally viable are held in memory and adapted responsive to newly arrived data, by changing their boundaries and locations. Accordingly, no raw data needs to be kept in memory, this is known as the online component of this type of algorithm. Subsequently, analytical descriptive clusters can be built offline on- demand based on such statistical summaries instead of using infinite raw data. A notable development here is CluStream [7] which can delete obsolete Micro-Clusters, browse through historical models, re-visiting historical models and "forgetting" obsolete concepts. . A shortcoming of CluStream is that only circular clusters can be built. DenStream algorithm [11] addresses this through the introduction of 'dense' Micro-Clusters to summarise clusters using an arbitrary shape. Another Micro-Cluster-based algorithm is ClusTree [18], which incrementally learns a dendrogram tree structure from Micro-Clusters.

4 Current Research in the Field of Data Stream Mining

Section 3 outlined some existing works developed over the last 20 years aiming to address the data stream analytics challenges. However, referring to the challenges and barriers as highlighted in Table 1, these are largely limited to overcoming Barrier 2. Here we summarise some recent work based on Micro-Clusters approach to address and overcome some of the remaining barriers included in 1.

4.1 Scalable (parallel) real-time Data Stream Mining Algorithms

Data Stream Mining algorithms are generally designed to be fast and scalable due to the essential requirement that they have to meet, namely the capability to adapt by a single pass through the data. However, this capability is often limited by the serial segments of the processing and little work has been carried out in the development of end-to-end-parallel DSM algorithms. In MC-NN [22] the authors proposed a parallel KNN-like algorithm. Here class-labelled Micro-Clusters are distributed over real-time setup of a MapReduce architecture and each Mapper (node in cluster) is to provide votes based on the Micro-Cluster distance with respect to newly arriving unlabelled data instances. MC-NN has performed well in terms of speed and accuracy in comparison with some of the fastest data stream classifiers. MC-NN has excelled when confronted with continuously changing data streams and has shown a very low latency to adapt to concept drift.

4.2 Parallel real-time Architectures

The works summarised in Section 4.1 have also developed suitable software architectures to enable the parallel real-time execution of the MC-NN and similar algorithms. This architecture uses a SAMZA [4] layer and Kafka [3] messaging system to achieve real-time instance-by-instance processing using the MapReduce framework over Hadoop [1] nodes. Here SAMZA enables synchronised real-time instance-by-instance processing of external messages over the Hadoop MapReduce framework. In this architecture SAMZA makes use of the low latency and high throughput publish and subscribe system of Kafka. With this parallel architecture MC-NN showed considerable speed-up over its serial version using up to 12 Hadoop nodes.

4.3 Real-time pre-processing Techniques

With respect to real-time pre-processing, very little work has been conducted and in practise often pre-processing techniques for batch data are applied. Whereas this may yield acceptable results in many cases, it does not take the ever-changing nature of data streams into account. For example, frequently used min-max normalisation can be applied in real-time but relies on stable min and max values. A recent work based on Micro-Clusters [15] aims to develop real-time feature selection techniques. In practise feature selection for data stream mining applications is applied once and then the chosen data stream mining algorithms continue to use these features. However, it is possible that the relevance of features change during concept drift and the original feature selection may not be no longer optimal. This work tracks Micro-Cluster rate of change (velocity) and the trajectory of Micro-Cluster movement in order to assess which features have been involved in a concept drift and to which extent. Based on this a real-time re-selection of the feature system was realised yielding improved accuracy over time in comparison with other methods which kept to the original feature selections.

5 Discussion and Conclusions

This article has defined 6 challenges and barriers to achieving true instance-by-instance data stream analytics in real-time (see Table 1). Subsequent sections have then outlined how established algorithms and current research aims to overcome these barriers, notably through Micro-Cluster-based approaches which have resulted in a variety of different kinds of analytics algorithms. However, challenges 4, 5 and 6 remain largely unaddressed. With respect to challenge 4, multi-modality of data sources, different time-stamped data from diverse heterogeneous data sources make it difficult to compose data frames as needed by data stream mining algorithms. Here some ad hoc and application-specific solutions have been implemented. However, a generic approach adaptable to a range of problems remains to be explored. With respect to challenge 5, class label sparsity, many predictive classification algorithms for data streams are supervised and thus require timely and frequent feedback about the ground truth for their predictions. Here some works have been prototyped based on ensemble approaches [17], however, as yet, there exists no viable solution for concrete applications. With respect to challenge 6, some ecosystems for parallel processing of streaming data in real-time exist, such as the Apache Flink project [2], however, their focus lies mainly on the parallel and in-memory processing of data in micro batches, which does not take full advantage of existing instance-by-instance data stream mining algorithms. Further, such systems often lack a full end-to-end architecture and focus more on the data processing efficiency. Here an end-to-end architecture should comprise components ranging from data ingestion and provenance-plausibility-aware data frame composition through to analytics algorithms / workflows and end-user dashboards to control, configure and examine the end-to-end data stream analytics pipeline.

References

1. Apache hadoop. `https://hadoop.apache.org/`. Accessed: 22-03-2020.
2. Apache hadoop. `https://flink.apache.org/`. Accessed: 22-03-2020.
3. Apache kafka. `https://kafka.apache.org/`. Accessed: 22-03-2020.
4. Apache SAMZA. `http://samza.apache.org/`. Accessed: 22-03-2020.
5. Data never sleeps 6.0. `https://www.domo.com/learn/data-never-sleeps-6`. Accessed: 22-03-2020.
6. IBM. `https://www.ibm.com/blogs/insights-on-business/consumer-products/2-5-quintillion-bytes-of-data-created-every-day-how-does-cpg-retail-manage-it/`. Accessed: 21-12-2010.
7. Charu C Aggarwal, S Yu Philip, Jiawei Han, and Jianyong Wang. A framework for clustering evolving data streams. In *Proceedings 2003 VLDB conference*, pages 81–92. Elsevier, 2003.
8. Brian Babcock, Shivnath Babu, Mayur Datar, Rajeev Motwani, and Jennifer Widom. Models and issues in data stream systems. In *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 1–16, 2002.
9. Manuel Baena-Garcıa, José del Campo-Ávila, Raúl Fidalgo, Albert Bifet, R Gavalda, and R Morales-Bueno. Early drift detection method. In *Fourth international workshop on knowledge discovery from data streams*, volume 6, pages 77–86, 2006.
10. Albert Bifet and Ricard Gavalda. Learning from time-changing data with adaptive windowing. In *Proceedings of the 2007 SIAM international conference on data mining*, pages 443–448. SIAM, 2007.
11. Feng Cao, Martin Estert, Weining Qian, and Aoying Zhou. Density-based clustering over an evolving data stream with noise. In *Proceedings of the 2006 SIAM international conference on data mining*, pages 328–339. SIAM, 2006.
12. Mayur Datar, Aristides Gionis, Piotr Indyk, and Rajeev Motwani. Maintaining stream statistics over sliding windows. *SIAM journal on computing*, 31(6):1794–1813, 2002.
13. Joao Gama and Petr Kosina. Learning decision rules from data streams. In *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.
14. Joao Gama, Pedro Medas, Gladys Castillo, and Pedro Rodrigues. Learning with drift detection. In *Brazilian symposium on artificial intelligence*, pages 286–295. Springer, 2004.
15. Mahmood Shakir Hammoodi, Frederic Stahl, and Atta Badii. Real-time feature selection technique with concept drift detection using adaptive micro-clusters for data stream mining. *Knowledge-Based Systems*, 161:205–239, 2018.
16. Geoff Hulten, Laurie Spencer, and Pedro Domingos. Mining time-changing data streams. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 97–106, 2001.

17. Mobin M Idrees, Leandro L Minku, Frederic Stahl, and Atta Badii. A heterogeneous online learning ensemble for non-stationary environments. *Knowledge-Based Systems*, 188:104983, 2020.
18. Philipp Kranen, Ira Assent, Corinna Baldauf, and Thomas Seidl. The clustree: indexing micro-clusters for anytime stream mining. *Knowledge and information systems*, 29(2):249–272, 2011.
19. Thien Le, Frederic Stahl, Mohamed Medhat Gaber, João Bártolo Gomes, and Giuseppe Di Fatta. On expressiveness and uncertainty awareness in rule-based classification for data streams. *Neurocomputing*, 265:127–141, 2017.
20. Ewan S Page. Continuous inspection schemes. *Biometrika*, 41(1/2):100–115, 1954.
21. Mark Tennant, Frederic Stahl, and Joao Bártolo Gomes. Fast adaptive real-time classification for data streams with concept drift. In *International Conference on Internet and Distributed Computing Systems*, pages 265–272. Springer, 2015.
22. Mark Tennant, Frederic Stahl, Omer Rana, and João Bártolo Gomes. Scalable real-time classification of data streams with concept drift. *Future Generation Computer Systems*, 75:187–199, 2017.