# An Approach to Real Time Mining of Big Data

Amrutanshu Panigrahi[1], Sunil Panigrahi[2] and Manmath Nath Dash[3]

[1,3]Associate Professor, Department of Computer Science Engineering, Gandhi Institute For Technology (GIFT), Bhubaneswar

[2]Assistant Professor, Department of Computer Science Engineering, Gandhi Engineering College, Bhubaneswar

Publishing Date: January 16, 2016

## Abstract

Streaming data analysis in real time is becoming the fastest and most efficient way to obtain useful knowledge from what is happening now, allowing organizations to react quickly when problems appear or to detect new trends helping to improve their performance. Evolving data streams are contributing to the growth of data created over the last few years. We are creating the same quantity of data every two days, as we created from the dawn of time up until 2003. Evolving data streams methods are becoming a low-cost, green methodology for real time online prediction and analysis. We discuss the current and future trends of mining evolving data streams, and the challenges that the field will have to overcome during the next years.

## 1 Introduction

Nowadays, the quantity of data that is created every two days is estimated to be 5 exabytes. This amount of data is similar to the amount of data created from the dawn of time up until 2003. Moreover, it was estimated that 2007 was the first year in which it was not possible to store all the data that we are producing. This massive amount of data opens new challenging discovery tasks.

Data stream real time analytics are needed to manage the data currently generated, at an ever increasing rate, from such applications as: sensor networks, measurements in network monitoring and traffic management, log records or click-streams in web exploring, manufacturing processes, call detail records, email, blogging, twitter posts and others [5]. In fact, all data generated can be considered as streaming data or as a snapshot of streaming data, since it is obtained from an interval of time.

In the data stream model, data arrive at high speed, and algorithms that process them must do so under very strict constraints of space and time. Consequently, data streams pose several challenges for data mining algorithm design. First, algorithms must make use of limited resources (time and memory). Second, they must deal with data whose nature or distribution changes over time.

We need to deal with resources in an efficient and low-cost way. Green computing is the study and practice of using computing resources efficiently. A main approach to green computing is based on algorithmic efficiency. In data stream mining, we are interested in three main dimensions:

- accuracy
- amount of space (computer memory) necessary
- the time required to learn from training examples and to predict

These dimensions are typically interdependent: adjusting the time and space used

by an algorithm can influence accuracy. By storing more pre-computed information, such as look up tables, an algorithm can run faster at the expense of space. An algorithm can also run faster by processing less information, either by stopping early or storing less, thus having less data to process. The more time an algorithm has, the more likely it is that accuracy can be increased.

The issue of the measurement of three evaluation dimensions simultaneously has led to another important issue in data stream mining, namely estimating the combined cost of performing the learning and prediction processes in terms of time and memory. As an example, several rental cost options exist:

- Cost per hour of usage: Amazon Elastic

  service that provides resizable compute capacity in the cloud. Cost depends on the time and on the machine rented (small instance with 1.7 GB, large with 7.5 GB or extra large with 15 GB).

Every GB of RAM deployed for 1 hour equals one RAM-Hour.

In [4, 2] the use of RAM-Hours was introduced as an evaluation measure of the resources used by streaming algorithms. Ev-ery GB of RAM deployed for 1 hour equals one RAM-Hour.

## 2 New problems:Struc-tured classification

A new important and challenging task may be the structured pattern classification problem. Patterns are elements of (possibly infinite) sets endowed with a partial order relation . Examples of patterns are itemsets, sequences, trees and graphs.

The structured pattern classification problem is defined as follows. A set of examples of the form (t, y) is given, where y is a discrete class label and t is a pattern. The goal is to produce from these examples a model $\hat{y} = f(t)$ that will predict the classes y of future pattern examples

Most standard classification methods can only deal with vector data, which is but one of many possible pattern structures. To apply them to other types of patterns, such as graphs, we can use the following approach: we convert the pattern classificationComputeCloud(AmazonEC2)isaweb problem into a vector classification learning task, transforming patterns into vectors of attributes. Each attribute denotes the presence or absence of particular subpatterns, and we create attributes for all frequent subpatterns, or for a subset of these.

- Cost per hour and memory used: As the number of frequent subpatterns GoGrid is a web service similar to Ama- may be very large, we may perform a feature zon EC2, but it charges by RAM-Hours. selection process, selecting a subset of these frequent subpatterns, maintaining exactly or approximately the same information.

The structured output classification problem is even more challenging and is defined as follows. A set of examples of the form (t, y) is given, where t and y are patterns. The goal is to produce from these examples a pattern model $\hat{y} = f(t)$ that will predict the patterns y of future pattern examples. A way to deal with a structured output classification problem is to convert it to a multi-label classification problem, where the output pattern y is converted into a set of labels representing a subset of its frequents subpatterns.

Therefore, data stream multi-label classification methods may offer a solution to the structured output classification problem.

## 3 New applications: social networks

A future trend in mining evolving data streams will be how to analyze data from social networks and micro-blogging applications such as Twitter. Micro-blogs and Twitter data follow the data stream model. Twitter data arrive at high speed, and algorithms that process them must do so under very strict constraints of space and time.

The main Twitter data stream that provides all messages from every user in real-time is called Firehose and was made available to developers in 2010. This streaming data opens new challenging knowledge discovery issues. In April 2010, Twitter had 106 million registered users, and 180 million unique visitors every month. New users were signing up at a rate of 300,000 per day. Twitter's search engine received around 600 million search queries per day, and Twitter received a total of 3 billion requests a day via its API. It could not be clearer in this application domain that to deal with this amount and rate of data, streaming techniques are needed.

Sentiment analysis can be cast as a classification problem where the task is to classify messages into two categories depending on whether they convey positive or negative feelings. See [8] for a survey of sentiment analysis, and [6] for opinion mining techniques.

To build classifiers for sentiment analysis, we need to collect training data so that we can apply appropriate learning algorithms. Labeling tweets manually as positive or negative is a laborious and expensive, if not impossible, task. However, a significant advantage of Twitter data is that many tweets have author-provided sentiment indicators: changing sentiment is implicit in the use of various types of emoticons. Smileys or emoticons are visual cues that are associated with emotional states. They are constructed using the characters available on a standard keyboard, representing a facial expression of emotion. Hence we may use these to label our training data.

When the author of a tweet uses an emoticon, they are annotating their own text with an emotional state. Such annotated tweets can be used to train a sentiment classifier [1, 3].

## 4 New techniques: Hadoop, S4 or Storm

A way to speed up the mining of streaming learners is to distribute the training process onto several machines. Hadoop MapReduce is a programming model and software framework for writing applications that rapidly process vast amounts of data in parallel on large clusters of compute nodes.

A MapReduce job divides the input dataset into independent subsets that are processed by map tasks in parallel. This step of mapping is then followed by a step of reducing tasks. These reduce tasks use the output of the maps to obtain the final result of the job.

Apache S4 [7] is a platform for processing continuous data streams. S4 is designed specifically for managing data streams. S4 apps are designed combining streams and processing elements in real time. Storm from Twitter uses a similar approach.

Ensemble learning classifiers are easier to scale and parallelize than single classifier methods. They are the first, most obvious, candidate methods to implement using parallel techniques.

## 5 Conclusions

We have discussed the challenges that in our opinion, mining evolving data streams will have to deal during the next years. We have outlined new areas for research. These include structured classification and associated application areas as social networks.

Our ability to handle many exabytes of data across many application areas in the future will be crucially dependent on the existence of a rich variety of datasets, techniques and software frameworks. There is no doubt that data stream mining offers many challenges and equally many opportunites as the quantity of data generated in real time is going to continue growing.

## References


[1] A. Bifet and E. Frank. Sentiment knowledge discovery in Twitter streaming data. In Proc 13th International Conference on Discovery Science, Canberra, Australia, pages 1–15. Springer, 2010.

[2] A. Bifet, G. Holmes, R. Kirkby, and B. Pfahringer. MOA: Massive Online Analysis http://moa.cms.waikato.ac.nz/. Journal of Machine Learning Research (JMLR), 2010.

[3] A. Bifet, G. Holmes, and B. Pfahringer. Moa-tweetreader: Real-time analysis in twitter streaming data. In Discovery Science, pages 46–60, 2011.

[4] A. Bifet, G. Holmes, B. Pfahringer, and E. Frank. Fast perceptron decision tree learning from evolving data streams. In PAKDD, 2010.

[5] J. Gama. Knowledge discovery from data streams. Chapman & Hall/CRC, 2010.

[6] B. Liu. Web data mining; Exploring hyperlinks, contents, and usage data. Springer, 2006.

[7] L. Neumeyer, B. Robbins, A. Nair, and A. Kesari. S4: Distributed stream computing platform. In ICDM Workshops, pages 170–177, 2010.

[8] B. Pang and L. Lee. Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval, 2(1-2):1–135, 2008.