Diego Garcia-Olano
DAKD FALL ‘13

Data mining of Data Streams:
An overview of the state of the art

Data Streams are continuously produced at high-speed in dynamic, time-changing environments. Mining data streams then is concerned with extracting knowledge structures represented in models and patterns in these non stopping streams of information[1]. These include data generated in real time via meteorological, satellite and astronomical observations, sensor networks, social networks, financial transactions and internet data such as traffic logs, user queries, and email amongst others. Such activity generate huge amounts of online data which grow at an unlimited rate and accordingly present challenges most of the classical methods in the literature of data mining were not necessarily designed to handle[2]. Mining of data streams requires quick automated analysis of incoming data too large to reside in memory, whose distribution may evolve over time, and which may only be stored temporarily and probably only partially then if at all on disk. In fact, much data stream mining literature assumes the data is of such a magnitude that it can not be stored, and will probably never be seen by humans[3] and thus necessitates the use of one-pass algorithms and other techniques that have been developed to save computational resources, increase speed and maintain adequate accuracy of ever adapting models. These are all necessary before we even begin to discuss the challenges and developing state of solutions and strategies involved in mining multiple distributed interdependent streams[4], which are essential to projects such as Smarter Cities[5], and the Internet Of Things, “the next generation of Internet which will contain trillions of nodes representing various objects from small ubiquitous sensor devices and handhelds to large web servers and supercomputer clusters” [6], amongst others. We provide an overview of the state of the art in data mining of data streams focusing on topics within single, and then multiple stream analysis including concepts, algorithms, notable research, technologies and future direction. Given that the field of data stream mining requires an understanding of machine learning, data mining and streams, a potentially daunting task for the uninitiated, this overview will try not to assume a great deal of prior knowledge.

Mining of Single Streams:

A typical stream data model assumes data arrives at a processing engine at a rate that makes it infeasible to store everything in active storage[8] and so the usual approaches for querying, clustering and prediction based on batch procedures cannot cope with the time, memory and space requirements necessary to predict and produce usable output at any time from models based on the streams. Stream mining thus needs decision models capable of incorporating new information as data arrives in real time; forgetting outdated information; detecting changes and adapting the decision models to reflect the most recent information while taking into account the problem of concept drift[9].

Concept drift is what occurs if the distribution of the data upon which a model is based changes over time, either in a sudden burst or gradually, thus potentially causing the model to be become inconsistent with respect to new data. It can take various forms including "feature drift" where the distribution of observable input data X has changed, "real concept drift" where the underlying relationship between input X and target y changes, decision boundary changes, a changing of the prior distribution or the arrival of new information, ie. new concepts/classes. Research into how to handle concept drift in the context of data streams continues, and depending on the use case, predictive modeling for a single learner - classifiers, rankers and regression - or ensemble methods, focuses predominantly on monitors of the evolution of the error rate of the system, other types of change detection methods such as variable windows and dynamic integration, and/or adaptation via fixed windows, instance weights, and adaptive fusion rules [4].

Data mining techniques for data streams largely attempt to correctly adapt traditional techniques in data mining including prediction tasks such as classification, regression, ranking and time series analysis, clustering, and frequent pattern mining to handle streams. Additionally, because of the constraints on number of passes at the incoming data, the processing time needed, and the amount of memory available for usage, while applying data mining solutions to data streams its crucial to leverage the tradeoff between time and accuracy by understanding that many times it may be more efficient and sufficiently acceptable to get an approximate answer to a problem rather than an exact solution. Most algorithms for processing streams thus involve summarization of the stream by filter, projection and estimation methods or by looking at some subset or variant of a fixed-length sliding "window" of the most recently arrived data consisting of the last n elements for some, typically large, stream and then querying over that window as if it were a relation in a database.[8] Methods in the former, including sampling, load shedding and sketching, contain drawbacks such as inapplicability to certain domains, namely anomaly detection while synopsis data structures and aggregation, also in the former, provide only approximate answers and do not perform well with highly fluctuating data distributions[3]. Sliding window methods are preferred due to these shortcomings and are more concerned with the analysis of most recent data streams computing detailed analysis on more recent data items with summarized versions of older ones. A good summary of the pro's and cons of these techniques has been elaborated[12].

Classification methods represent the set of supervised learning techniques where a set of dependent variables need to be predicted based on another set of input attributes.
For regression, output is a numeric value whereas with classifiers output is assignment to a categorical or binary attribute. The process is divided into two phases; a model building one where a learning algorithm runs over a dataset to induce and train model which could be used in estimating an output, and then the model testing phase where the quality of the model is

evaluated over a test data set. Decision Trees, and Rule Based methods are two such algorithms used in inducing a model. These algorithms were originally designed to build classification models from static data sets where several passes over the stored data is possible, and not in the case of data streams, where it is necessary to process the entire data set in one pass. For the past decade then, there has been a good deal of research towards developing classification methods for streams which took on the handling of concept drift, by deciding which part of the stream to utilize for accuracy purposes, efficiency concerns, since the process of building and updating classifiers in the case of high speed data input is computationally complex, and finally robustness, to avoid the problem of overfitting, modeling random error or noise instead of the underlying relationship. Two techniques currently at the fore-front of stream mining are Two-phase techniques, also known as on demand classification, and Hoeffding bound-based techniques, including Very Fast Decision Trees, amongst other [10]. Two phase techniques contain an online component that stores summary statistics about the data streams and an offline one that performs clustering on the summarized data according to a number of user preferences such as the time frame and the number of clusters. On-demand classification then uses the clustering results to classify data using statistics of class distribution with the main motivation being that it be used over a time horizon which depends on the nature of the concept drift, smaller drifts requiring larger time horizons and vise versa. The summary statistics are represented in the form of class-label specific micro-clusters where all points in a cluster have the same class label, and at any given moment in time, the current set of micro-clusters can be used to perform the classification. Hoeffding bound-based techniques are generic strategies for scaling up machine learning algorithms by determining an upper bound for the learner's accuracy loss as a function of the number of examples/data records in each step of the algorithm. Thus a tradeoff exists between model accuracy, still an ongoing issue in the field, and the high speed and bounded memory of the technique. There exist many variants of very fast machine learning (VFML) techniques including very fast decision tree classification (VFDT) and very fast k-means clustering (VFKM) which itself has been extended to address specifically the problem of concept drift in evolving data streams by Hulten et al.[11]

Regression methods, similar to classification methods with the exception of outputting numeric values as opposed to categorical or binary attributes, are currently adapted for stream contexts including by adapting a similar method for Hoeffding-Based Regression Trees[13], and by dynamic integration of regression models.[14]

Large volumes of arriving data makes the traditional clustering algorithms inefficient, if we are for instance to consider calculating pairwise distances, and the quality of the clusters becomes poor when the underlying data evolves over time. The goal of clustering in streams is thus to continuously maintain a clustering structure in evolving time series data streams with single pass processing in a space efficient manner. In addition to the aforementioned very fast k-means clustering algorithm, and Clu-Stream which introduced the microclusters idea used in on-demand classification, there exist a wide variety of other clustering methods largely based on density search including DenStream and HPStream, and other probabilistic mixture models.[15]

Frequent Pattern Mining is another area of traditional data mining that has been adapted to the context of streams having uses in many important areas including market basket analysis, intrusion detection, churn prediction, feature selection, query analysis, and clickstream analysis, and anomaly detection. Specifically, the task of frequent pattern mining is to find the item sets that appear frequently from a data stream, an item set being any subset of the set of all items. The vast majority of stream pattern mining algorithms build and update a pattern lattice which is a batch summary of the incoming stream.[4] One particular algorithm CloStream mines only frequent closed itemsets,ie frequencies of distinct groupings which appear and thus making subsets redundant, by maintaining a hashtable of them, and a lookup list of identifiers to and supersets of each row of the hash table[16]

In summary, the main modeling principles of single stream mining include variations of “window” approaches to take into account the most recent data, adaptation of models to be incrementally updatable and the use of combinations of models built for different situations and concepts.

Ref [4] provides good outlook on the current state of the field and future research directions. Currently many solutions are available for only “basic” settings, implying the need to expand and test in varied, rigorous ways. Additionally, the field as a whole needs systemization of settings and terminology, and lacks real-world benchmarks and tasks, though it appears the latter point is beginning to get dealt with.[20]

Mining of Multiple Streams:

Coordination of sensor signals, stock trading, air quality monitoring and real-time user activity on social networks are all applications of mining over multiple streams. Stream mining may concern streams which are correlated in unknown ways, are interdependent - depending on one another, and/or distributed. Mining multiple streams implies, aligning them, combining them – not least for error correction, learning a model from them and adapting to drift.[4] It is a difficult combination and as of yet has not been well solved, but it will most surely involve the advancement of algorithms which speed up search and mining tasks significantly.[7] The two main areas of focus currently are on mining distributed and multiple interdependent streams. Overall distributed stream mining focuses on communication among and querying over distributed streams and learning on sensor networks whereas mining multiple interdependent streams is concerned with the interplay between learning models of the individual streams and on modeling entities that are enriched with stream data.

As distributed systems have additional requirements of needing to enforce small process time and reduced communication amongst nodes, sensors for most use cases, much research has gone into the development of monitoring algorithms. A novel one involves a geometric approach by which an arbitrary global monitoring task can be split into a set of constraints applied locally on each of the streams which are used to locally filter out data increments that do not affect the monitoring outcome, thus avoiding unnecessary communication to enables monitoring of arbitrary threshold functions for feature selection over distributed data streams in an efficient manner.[18] Similarly algorithms have been developed to analyze multiple information sources where data can be gigantic, noisy, unreliable, dynamically evolving, highly imbalanced, and heterogeneous and learn of their correlations by explore their similarities (consensus combination), or their differences (inconsistency detection) for classification and anomaly detection.[17]

For clustering distributed sources of data streams, with the goal of minimizing communication and computational cost while ensuring accuracy of the clustering, its necessary to continuously maintain a cluster structure of the sensors producing data and to use a conquer and divide approach to design efficient methods to cope with dynamic data, and effectively perform clustering to yield clusters of comparable quality to a centralized clustering. Its been shown that by spreading that responsibility to the nodes distributed in the network, and by using K-center clustering, along with a suite of algorithms that vary based on which centralized algorithm they derive from and whether they maintain a single global clustering or many local clusterings that can be merged together, such requirements can be met[19]. A good overview of the generalized process along with a space saving algorithm for monitoring states and an explanation of the benefits in lowering stress on the overall system yielded by online discretization of individual sensors, and the monitoring and online clustering of frequent states can be found [4].

Two example situations arising in the mining of interdependent streams help illustrate current practices within the specific discipline; one where the streams are concurrent and respond to external, a priori unknown events, such as with topic evolution and detection of bursty events, and another where the streams emanate from a population of fixed entities, such as is the case with risk prediction and consumer change in attitude towards products.[4] In the first, we mine the streams together to acquire a richer understanding or obtain earlier indicators of these events by learning and adapting a model on the stream instances themselves. Advances in pattern discovery, classification and clustering towards this sort of problem within interdependent streams includes research on the use of frequent itemset discovery to predict the values of health parameters as streams of patients as they move, combining streams of different types of sensor recordings for event detection and tracking where the events of interest associated to the emergence of cyclones, and the study of asynchronous text streams to extract a common set of topics by using word distributions iteratively to bring the streams in sync and to refine the topics found thus far.[21][22][23] The second situation by contrast is mined to acquire insights on the behavior of the fixed entities by learning and adapting a model on entities and not the streams themselves as in the first case. This circumstance brings forth two challenges; synchronization, knowing which instances to exploit and which entities to account for adoption, and aggregation, namely how to aggregate and embed information from instances into a relational entity for exploitation.

Another area of practical importance to the mining of multiple streams, which includes mobile phones and ubiquitous streams, is the adoption granularity-based techniques which adapt mining techniques to change their resource consumption patterns over time according to availability of resources thus making them resource aware which could be exploited over distributed networks.[10]

Conclusion:

Two IBM initiatives, both under their Smart Planet proposal, which will be major fronts for the implementation of single and multiple stream mining techniques in the near future are Smarter Cities especially as it relates to MegaCities (NY, London, etc) and the Internet Of Things (IoT). These are of particular interest because what will distinguish the production of their data from earlier ones is the wider scale automatic adoption of them via feeds for public systems and personal uses in our physical daily lives.

Technologies:

Data Stream Mining technologies

- MOA - open source framework for data stream mining. written in java
- VFML - open source toolkit for mining high-speed data streams and very large data sets. written in C from Domingos and Hulten
- Rapid Miner data stream plugin
- Storm - open source distributed realtime computation system to reliably process unbounded streams of data, doing for realtime processing what Hadoop did for batch processing.
- S4 - general-purpose, distributed, scalable, fault-tolerant, pluggable platform that allows programmers to easily develop applications for processing continuous, unbounded streams of data

Large Scale Machine Learning

- Mahout - Apache machine learning library
- Vowpal Wabbit - fast, scalable,useful learning algorithm run by microsoft
- Apache Samza - a distributed stream processing framework that uses Apache Kafka for messaging, and Apache Hadoop YARN to provide fault tolerance, processor isolation, security, and resource management.
- Samoa - a platform for online mining of big data streams in a cluster/cloud environment featuring a pluggable architecture that allows it to run on several distributed stream processing engines such as S4 and Storm

Acknowledgements:

I would like to thank Professor Ricard Gavalda for his guidance towards informative relevant papers, discussing the high level generalities of mining of data streams and for answering questions from an overly needy grad student.

Bibliography :

[1] Mohamed Medhat Gaber, Arkady B. Zaslavsky, Shonali Krishnaswamy: Mining data streams: a review. SIGMOD Record 34(2): 18-26 (2005)

[2] C. Aggarwal. A Survey of Stream Clustering Algorithms. In Data Clustering: Algorithms and Applications, CRC Press, 2013

[3] Joao Gama, Mohamed Medhat Gaber, State-of-the-Art in Data Stream Mining. Tutorial presented at ECML & PKDD 2007.

[4] Albert Bifet, João Gama, Ricard Gavaldà, Georg Krempl, Mykola Pechenizkiy, Bernhard Pfahringer, Myra Spiliopoulou, Indrė Žliobaitė, Advanced Topics on Data Stream Mining. Tutorial presented at ECML PKDD 2012. available at https://sites.google.com/site/advancedstreamingtutorial/

[5] Freddy Lecue, Spyros Kotoulas, Pol Mac Aonghusa , Capturing The Pulse of Cities: Opportunity and Research Challenges for Robust Stream Data Reasoning In Proceedings of the 1st AAAI 2012 Workshop on Semantic Cities, AAAI Press 2012

[6] Shen Bin; Liu Yuan; Wang Xiaoyi, "Research on data mining models for the internet of things," *Image Analysis and Signal Processing (IASP), 2010 International Conference on* , vol., no., pp.127,132, 9-11 April 2010 doi: 10.1109/IASP.2010.5476146

[7] Thanawin Rakthanmanon, Bilson J. L. Campana, Abdullah Mueen, Gustavo E. A. P. A. Batista, M. Brandon Westover, Qiang Zhu, Jesin Zakaria, Eamonn J. Keogh: Searching and mining trillions of time series subsequences under dynamic time warping. KDD 2012: 262-270

[8] Chapter 4: Mining Data Streams. from Mining of massive datasets, Anand Rajaraman, and Jeffrey David Ullman. Cambridge University Press, Cambridge, (2012)

[9] Alexey Tsymbal, The Problem of Concept Drift: Definitions and Related Work, 2004

[10] Mohamed Medhat Gaber. Advances in data stream mining. WIREs Data Mining Knowl Discov 2012, 2: 79–85 doi: 10.1002/widm.52

[11] Geoff Hulten, Laurie Spencer, Pedro Domingos: Mining time-changing data streams. KDD 2001: 97-106

[12] Mohamed Medhat Gaber, Arkady B. Zaslavsky, Shonali Krishnaswamy: A Survey of Classification Methods in Data Streams. Data Streams - Models and Algorithms 2007:39-59

[13] Elena Ikonomovska, João Gama, Bernard Zenko, Saso Dzeroski: Speeding-Up Hoeffding-Based Regression Trees With Options. ICML 2011: 537-544

[14] Niall Rooney, David W. Patterson, Sarab S. Anand, Alexey Tsymbal: Dynamic Integration of Regression Models. Multiple Classifier Systems 2004: 164-173

[15] C. Aggarwal. A Survey of Stream Clustering Algorithms, In "Data Clustering: Algorithms and Applications", ed. C. Aggarwal and C. Reddy, CRC Press, 2013.

[16] Show-Jane Yen, Yue-Shi Lee, Cheng-Wei Wu, Chin-Lin Lin: An Efficient Algorithm for Maintaining Frequent Closed Itemsets over Data Stream. IEA/AIE 2009: 767-776.

[17] Gao, 2011] Jing Gao, Exploring the Power of Heterogenous Information Sources, PhD Thesis, Ch. 5 "Consensus Combination for Stream Classification ", Univ. Illinois, Urbana-Champaign 2011

[18] I. Sharfman, A. Schuster, D. Keren, A Geometric Approach to Monitoring Distributed

DataStreams, SIGMOD 2006
[19] Graham Cormode, S. Muthukrishnan, Wei Zhuang: Conquering the Divide: Continuous Clustering of Distributed Data Streams. 1036-1045
[20] João Gama, Raquel Sebastião, and Pedro Pereira Rodrigues. 2013. On evaluating stream learning algorithms. *Mach. Learn.* 90, 3 (March 2013), 317-346.
DOI=10.1007/s10994-012-5320-9 http://dx.doi.org/10.1007/s10994-012-5320-9
[21] M. Hassani, T. Seidl "Towards a Mobile Health Context Prediction: Sequential Pattern Mining in Multiple Streams", IEEE Int. Conf. on Mobile Data Management 2011
[22] A. Talukder "Event Data Mining and Classification from Multiple Streaming Sources", Workshops of ICDM 2010
[23] Xiang Wang, Kai Zhang, Xiaoming Jin, Dou Shen, "Mining Common Topics from Multiple Asynchronous Text Streams" WSDM 2009