SAMOA: A Platform for Mining Big Data Streams

Gianmarco De Francisci Morales
Yahoo! Research
Barcelona, Spain
gdfm@yahoo-inc.com

ABSTRACT

Social media and user generated content are causing an ever growing data deluge. The rate at which we produce data is growing steadily, thus creating larger and larger streams of continuously evolving data. Online news, micro-blogs, search queries are just a few examples of these continuous streams of user activities. The value of these streams relies in their freshness and relatedness to ongoing events. However, current (de-facto standard) solutions for big data analysis are not designed to deal with evolving streams.

In this talk, we offer a sneak preview of SAMOA, an upcoming platform for mining big data streams. SAMOA is a platform for online mining in a cluster/cloud environment. It features a pluggable architecture that allows it to run on several distributed stream processing engines such as S4 and Storm. SAMOA includes algorithms for the most common machine learning tasks such as classification and clustering. Finally, SAMOA will soon be open sourced in order to foster collaboration and research on big data stream mining.

Categories and Subject Descriptors

H.2.8 [**Information Systems Applications**]: Database Applications—*Data mining*; H.3.4 [**Information Systems**]: Systems and Software—*Distributed systems*

General Terms

Performance, Algorithms, Design

Keywords

Big Data, Data Streams, Stream Mining, Distributed Computing, Machine Learning, Open Source

1. INTRODUCTION

Big data is data whose characteristics forces us to look beyond the traditional methods that are prevalent at the time. Currently, there are two main ways to deal with big data: streaming algorithms and distributed computing. We argue that neither alone is sufficient to satisfy the near-future needs for big data stream mining, rather a combination of the two approaches under an open source umbrella is needed.

WWW 2013 Companion, May 13–17, 2013, Rio de Janeiro, Brazil.
ACM 978-1-4503-2038-2/13/05.

1.1 Streaming is not enough

Most data generated is originally streaming data. This fact is especially true for data representing measurements, actions and interactions, such as the one coming from sensor networks or the Web. In-rest data is just a snapshot of streaming data obtained from an interval of time.

In the streaming model, data arrives at high speed, and algorithms must process it in one pass under very strict constraints of space and time. Streaming algorithms use probabilistic data structures in algorithm give fast, approximated answers. However, sequential online algorithms are limited by the memory and bandwidth of a single machine.

Achieving results faster and scaling to larger data streams requires to resort to parallel and distributed computing. MapReduce [5] is currently the de-facto standard programming paradigm in this area, mostly thanks to the popularity of Hadoop[1], an open source implementation of MapReduce started at Yahoo!.

1.2 The Elephant in the Room

Hadoop and big data have come to become synonyms in the last years. Hadoop has evolved from a simple open source clone of MapReduce to a flourishing ecosystem of related projects for data storage, management, representation, processing and analysis.

MapReduce and streaming are two fundamentally different programming paradigms, albeit related from a theoretical point of view [6]. In recent years, mostly because of the popularity of Hadoop, there have been various attempts to shoehorn streaming and incremental computation on top of MapReduce, such as Hadoop Online Prototype [4], HaLoop [3], and DEDUCE [9]. However, all these systems are adaptations and hybridizations rather than principled approaches, and therefore present limited support for proper streaming computation.

1.3 Big Data Streams: Volume + Velocity

Big data is often understood along 3 dimensions, called 3 V's: Volume, Variety and Velocity [7]. Big data streams are characterized by having high volume and high velocity. Additionally, when dealing with web and social streams variety is given for granted.

The key to deal with such complex data is, in our opinion, to combine streaming with distributed computing and open source. The streaming paradigm is necessary to deal with the velocity of the data, distributed computing to deal with the volume of the data, and being open source for the

[1] `http://hadoop.apache.org`

variety. While the first two points are easy to understand, the latter deserves some explanation. No two companies or individuals have the same needs, and open source is a guarantee of openness and adaptability. Indeed, with an open source solution any skilled person can modify the code to suit their needs.

For the reasons above, we exclude from our consideration existing commercial SPEs such as IBM InfoSphere Streams[2], Microsoft StreamInsight[3], and StreamBase[4].

Open source, distributed Stream Processing Engines (SPEs) have been the focus of much research and development recently. One of the first ones to be available as open source was Borealis [1]. Currently, S4 [10] and Storm[5] are the state-of-the-art. They draw inspiration from both traditional SPEs and MapReduce. Similarly to MapReduce, in these systems data is seen as a sequence of records which can be routed to the right processing element based on the value of a key. Similarly to SPEs, records are processed one by one, and any aggregation (such as the reduce phase in MapReduce) is left to the user.

2. SAMOA

While the computing infrastructure is available, algorithms and tools for big data stream mining and machine learning are lacking. Weka [8] and R[6] are the traditional tools used for machine learning. Mahout[7] is a scalable learning library for Hadoop. MOA [2] is a framework for data stream mining, albeit for single machines.

SAMOA (Scalable Advanced Massive Online Analysis) aims to fill this gap, as summarized by Figure 1. SAMOA is both a platform and a library of algorithms for big data stream mining and machine learning. It features a pluggable architecture that allows to run on top of several distributed SPEs. This capability is achieved by designing a minimal API that captures the essence of modern distributed SPEs. Initial support is provided for S4 and Storm, but bindings for new systems can be added easily. SAMOA takes care of hiding all the differences of the underlying SPEs in terms of API, model of computation and deployment.

SAMOA supports the most common machine learning tasks such as classification and clustering. It includes distributed versions of classical streaming algorithms such as Hoeffding decision trees and k-means-based clustering. SAMOA also provides a simplified API for the algorithm developer. Our hope is that other people will be able to use the API to implement distributed streaming algorithms easily. We also plan to add other types of tasks in the future, such as topic modeling and collaborative filtering.

Finally, SAMOA will be released as open source software under the Apache license 2.0.

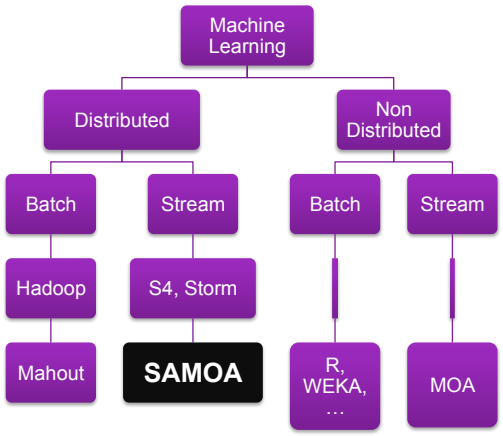

Figure 1: Taxonomy of machine learning tools.

3. ACKNOWLEDGMENTS

I thank my colleague Albert Bifet for having produced much of the material for this talk. My gratitude also goes to the rest of the people who have contributed to SAMOA: Matthieu Morel, Arinto Murdopo and Antonio Severien.

4. REFERENCES

[1] D. J. Abadi, Y. Ahmad, M. Balazinska, M. Cherniack, J.-h. Hwang, W. Lindner, A. S. Maskey, E. Rasin, E. Ryvkina, N. Tatbul, Y. Xing, and S. Zdonik. The Design of the Borealis Stream Processing Engine. In *CIDR '05: 1st Conference on Innovative Data Systems Research*, pages 277–289, 2005.

[2] A. Bifet, G. Holmes, R. Kirkby, and B. Pfahringer. MOA: Massive Online Analysis. *The Journal of Machine Learning Research*, 11:1601–1604, 2010.

[3] Y. Bu, B. Howe, M. Balazinska, and M. D. Ernst. HaLoop: efficient iterative data processing on large clusters. *VLDB Endowment*, 3(1-2):285–296, 2010.

[4] T. Condie, N. Conway, P. Alvaro, J. M. Hellerstein, K. Elmeleegy, and R. Sears. MapReduce Online. Technical Report UCB/EECS-2009-136, University of California, Berkeley, 2009.

[5] J. Dean and S. Ghemawat. MapReduce: Simplified Data processing on Large Clusters. In *OSDI '04: 6th Symposium on Opearting Systems Design and Implementation*, pages 137–150. USENIX Association, 2004.

[6] J. Feldman, S. Muthukrishnan, A. Sidiropoulos, C. Stein, and Z. Svitkina. On distributing symmetric streaming computations. *ACM Transactions on Algorithms*, 6(4): 1–19, 2010.

[7] Gartner. Gartner Says Solving 'Big Data' Challenge Involves More Than Just Managing Volumes of Data, 2011. URL `http://www.gartner.com/it/page.jsp?id=1731916`.

[8] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The WEKA data mining software. *SIGKDD Explorations*, 11(1):10, 2009.

[9] V. Kumar, H. Andrade, B. Gedik, and K.-L. Wu. DEDUCE: At the Intersection of MapReduce and Stream Processing. In *EDBT '10: 13th International Conference on Extending Database Technology*, pages 657–662. ACM Press, 2010.

[10] L. Neumeyer, B. Robbins, A. Nair, and A. Kesari. S4: Distributed Stream Computing Platform. In *ICDMW '10: 10th International Conference on Data Mining Workshops*, pages 170–177. IEEE, 2010.

[2] `http://www.ibm.com/software/data/infosphere/streams`
[3] `http://www.microsoft.com/en-us/sqlserver/solutions-technologies/business-intelligence/streaming-data.aspx`
[4] `http://www.streambase.com`
[5] `http://storm-project.net`
[6] `http://www.r-project.org`
[7] `http://mahout.apache.org`