Journal of Machine Learning Research 16 (2015) 149-153 Submitted 6/14; Revised 9/14; Published 1/15

SAMOA: Scalable Advanced Massive Online Analysis

Gianmarco De Francisci Morales GDFM@APACHE.ORG
Albert Bifet ABIFET@YAHOO.COM
Yahoo Labs
Av. Diagonal 177, 8th floor, 08018, Barcelona, Spain

Editor: Geoff Holmes

Abstract

SAMOA (SCALABLE ADVANCED MASSIVE ONLINE ANALYSIS) is a platform for mining big data streams. It provides a collection of distributed streaming algorithms for the most common data mining and machine learning tasks such as classification, clustering, and regression, as well as programming abstractions to develop new algorithms. It features a pluggable architecture that allows it to run on several distributed stream processing engines such as Storm, S4, and Samza. SAMOA is written in Java, is open source, and is available at `http://samoa-project.net` under the Apache Software License version 2.0.

Keywords: data streams, distributed systems, classification, clustering, regression, toolbox, machine learning

1. Introduction

Big data is "data whose characteristics forces us to look beyond the traditional methods that are prevalent at the time" (Jacobs, 2009). Currently, there are two main ways to deal with big data: streaming algorithms and distributed computing (e.g., MapReduce). SAMOA aims at satisfying the future needs for big data stream mining by combining the two approaches in a single platform under an open source umbrella.

Data mining and machine learning are well established techniques among web companies and startups. Spam detection, personalization, and recommendation are just a few of the applications made possible by mining the huge quantity of data available nowadays.

The usual pipeline for mining and modeling data (what "data scientists" do) involves taking a sample from production data, cleaning and preprocessing it to make it amenable to modeling, training a model for the task at hand, and finally deploying it to production. The final output of this process is a pipeline that needs to run (and be maintained) periodically in order to keep the model up to date.

In order to cope with web-scale data sets, data scientists have resorted to *parallel and distributed computing*. MapReduce (Dean and Ghemawat, 2004) 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. Hadoop and its ecosystem (e.g., Mahout[2]) have proven to be an extremely successful platform to support the aforementioned process at web scale.

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

[2] See `http://mahout.apache.org`

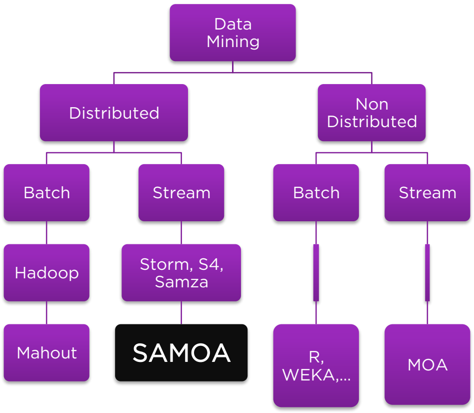

Figure 1: Taxonomy of data mining tools.

However, nowadays most data is generated in the form of a stream. Batch data is just a snapshot of streaming data obtained in an interval of time. Researchers have conceptualized and abstracted this setting in the *streaming model*. In this model data arrives at high speed, one instance at a time, and algorithms must process it in one pass under very strict constraints of space and time. Streaming algorithms make use of probabilistic guarantees to give fast approximated answers.

On the one hand, MapReduce is not suited to express streaming algorithms. On the other hand, traditional sequential online algorithms are limited by the memory and bandwidth of a single machine. *Distributed stream processing engines* (DSPEs) are a new emergent family of MapReduce-inspired technologies that address this issue. These engines allow to express parallel computation on streams, and combine the scalability of distributed processing with the efficiency of streaming algorithms. Examples of these engines include Storm,[3] S4,[4] and Samza.[5]

Alas, currently there is no common solution for mining big data streams, that is, for executing data mining and machine learning algorithms on a distributed stream processing engine. The goal of SAMOA is to fill this gap, as exemplified by Figure 1.

2. Description

SAMOA (Scalable Advanced Massive Online Analysis) is a platform for mining big data streams (De Francisci Morales, 2013). For a simple analogy, think of SAMOA as Mahout for streaming. As most of the rest of the big data ecosystem, it is written in Java.

SAMOA is both a framework and a library. As a framework, it allows algorithm developers to abstract from the underlying execution engine, and therefore reuse their code on different engines. It features a pluggable architecture that allows it to run on several distributed stream processing engines such as Storm, S4, and Samza. This capability is achieved by

[3] See `http://storm.apache.org`
[4] See `http://incubator.apache.org/s4`
[5] See `http://samza.incubator.apache.org`

designing a minimal API that captures the essence of modern DSPEs. This API also allows to easily write new bindings to port SAMOA to new execution engines. SAMOA takes care of hiding the differences of the underlying DSPEs in terms of API and deployment.

As a library, SAMOA contains implementations of state-of-the-art algorithms for distributed machine learning on streams. For classification, SAMOA provides the Vertical Hoeffding Tree (VHT), a distributed version of a streaming decision tree (Domingos and Hulten, 2000). For clustering, it includes an algorithm based on CluStream (Aggarwal et al., 2003). For regression, a decision rule learner (Thu Vu et al., 2014). The library also includes meta-algorithms such as bagging and boosting.

The platform is intended to be useful in both research and real world deployments.

3. Architecture

An algorithm in SAMOA is represented by a directed graph of nodes that communicate via messages along streams which connect pairs of nodes. Borrowing the terminology from Storm, this graph is called a *Topology*. Each node in a Topology is a *Processor* that sends messages through a *Stream*. A Processor is a container for the code implementing the algorithm. A Stream can have a single source but several destinations (akin to a pub-sub system). A Topology is built by using a *TopologyBuilder*, which connects the various pieces of user code to the platform code and performs the necessary bookkeeping in the background. The following is a code snippet to build a topology that joins two data streams in SAMOA.

```
TopologyBuilder builder = new TopologyBuilder();
Processor sourceOne = new SourceProcessor();
builder.addProcessor(sourceOne);
Stream streamOne = builder.createStream(sourceOne);

Processor sourceTwo = new SourceProcessor();
builder.addProcessor(sourceTwo);
Stream streamTwo = builder.createStream(sourceTwo);

Processor join = new JoinProcessor();
builder.addProcessor(join).connectInputShuffle(streamOne)
    .connectInputKey(streamTwo);
```

4. Machine Learning Algorithms

The Vertical Hoeffding Tree (VHT) is a distributed extension of the VFDT (Domingos and Hulten, 2000). The VHT uses vertical parallelism to split the workload across several machines. Vertical parallelism leverages the parallelism across attributes in the same example, rather than across different examples in the stream. In practice, each training example is routed through the tree model to a leaf. There, the example is split into its constituting attributes, and each attribute is sent to a different Processor instance that keeps track of sufficient statistics. This architecture has two main advantages over one based on horizontal parallelism. First, attribute counters are not replicated across several machines, thus reducing the memory footprint. Second, the computation of the fitness of an attribute for a split decision (via, e.g., entropy or information gain) can be performed in parallel. The drawback is that in order to get good performance, there must be sufficient inherent parallelism in the data. That is, the VHT works best for sparse data (e.g, bag-of-words models).

SAMOA includes a distributed version of CluStream, an algorithm for clustering evolving data streams. CluStream keeps a small summary of the data received so far by computing micro-clusters online. These micro-clusters are further refined to create macro-clusters by a micro-batch process, which is triggered periodically. The period can be configured via a command line parameter (e.g., every 10 000 examples).

SAMOA also includes adaptive implementations of ensemble methods such as bagging and boosting. These methods include state-of-the-art change detectors such as as ADWIN (Bifet and Gavaldà, 2007), DDM (Gama et al., 2004), EDDM (Baena-García et al., 2006), and Page-Hinckley (Gama et al., 2014). These meta-algorithms are most useful in conjunction with external single-machine classifiers which can be plugged in SAMOA. For instance, connectors for MOA (Bifet et al., 2010) are provided by the SAMOA-MOA package.[6]

The following listing shows how to download, build and run SAMOA.

```
# download and build SAMOA
git clone git@github.com:yahoo/samoa.git
cd samoa
mvn package

# download the Forest Cover Type data set
wget "http://downloads.sourceforge.net/project/moa-datastream/Datasets/
   Classification/covtypeNorm.arff.zip"
unzip "covtypeNorm.arff.zip"

# run SAMOA in local mode
bin/samoa local target/SAMOA-Local-0.2.0-SNAPSHOT.jar "PrequentialEvaluation
    -l classifiers.ensemble.Bagging -s (ArffFileStream -f covtypeNorm.arff)
    -f 100000"
```

5. Conclusions

SAMOA is a platform for mining big data streams. It supports the most common machine learning tasks such as classification, clustering, and regression. It also provides an API for algorithm developers that simplifies implementing distributed streaming algorithms.

SAMOA can be found at http://www.samoa-project.net/. The website includes a wiki, an API reference, and a developer's manual. Several examples of how the software can be used are available. The code is hosted on GitHub. SAMOA contains a test suite that is run on each commit on the GitHub repository via a continuous integration server.[7] Finally, SAMOA is released as open source software under the Apache Software License version 2.0.

We are grateful to all the people who contributed to SAMOA,[8] without whom the project could not have existed. We also thank Yahoo Labs Barcelona and its Web Mining group for the great support during the development of the project.

[6] See https://github.com/samoa-moa/samoa-moa
[7] See https://travis-ci.org/yahoo/samoa
[8] See http://samoa-project.net/contributors.html

References

Charu C. Aggarwal, Jiawei Han, Jianyong Wang, and Philip S. Yu. A framework for clustering evolving data streams. In *VLDB '03: 29th International Conference on Very Large Data Bases*, pages 81–92, 2003.

Manuel Baena-García, José del Campo-Ávila, Raúl Fidalgo, Albert Bifet, Ricard Gavaldá, and Rafael Morales-Bueno. Early drift detection method. In *IWKDDS '06: 4th International Workshop on Knowledge Discovery from Data Streams*, 2006.

Albert Bifet and Ricard Gavaldà. Learning from time-changing data with adaptive windowing. In *SDM '07: Seventh SIAM International Conference on Data Mining*, pages 443–448, 2007.

Albert Bifet, Geoff Holmes, Richard Kirkby, and Bernhard Pfahringer. MOA: Massive Online Analysis. *Journal of Machine Learning Research*, 11:1601–1604, 2010.

Gianmarco De Francisci Morales. SAMOA: A platform for mining big data streams. In *RAMSS '13: 2nd International Workshop on Real-Time Analysis and Mining of Social Streams @WWW '13*, 2013.

Jeffrey Dean and Sanjay Ghemawat. Mapreduce: Simplified data processing on large clusters. In *OSDI '04: 6th Symposium on Operating Systems Design and Implementation*, pages 137–150. USENIX Association, 2004.

Pedro Domingos and Geoff Hulten. Mining high-speed data streams. In *KDD '00: 6th International Conference on Knowledge Discovery and Data Mining*, pages 71–80, 2000.

João Gama, Pedro Medas, Gladys Castillo, and Pedro Pereira Rodrigues. Learning with drift detection. In *SBIA '04: 17th Brazilian Symposium on Artificial Intelligence*, pages 286–295, 2004.

João Gama, Indre Zliobaite, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. A survey on concept drift adaptation. *ACM Computing Surveys*, 46(4), 2014.

Adam Jacobs. The pathologies of big data. *Communications of the ACM*, 52(8):36–44, August 2009.

Anh Thu Vu, Gianmarco De Francisci Morales, João Gama, and Albert Bifet. Distributed adaptive model rules for mining big data streams. In *BigData '14: Second IEEE International Conference on Big Data*, 2014.