Roadmap

Triggers

Many algorithms in SAMOA have some action triggered by some condition (e.g., call this function every 1000 events). Some of the engines we run on could benefit from having this behavior exposed, in order to optimize their execution (e.g., by using windowing semantics). One such example is Apache Flink.

The goal of this project is to design an API for event triggering that can be used at the ML level to describe actions to be taken upon conditions, and exposed at the System level in order to be available for optimization by the underlying execution engine.

Distributed Naive Bayes

Feature Extraction Pipeline

Stochastic Gradient Descent in Java

Stochastic Gradient Descent (SGD) is a classical optimization algorithm used in many machine learning algorithms. Currently, there are some parallel implementations of the algorithm but they have two shortcomings: either they are in C or C++ or they work only for shared memory systems (or both) [1,2,3]. In any case, they are unsuitable to the modern big data ecosystem.

There are a few implementations in MapReduce, however MapReduce and Hadoop are not suitable to deal with streams of data. The goal of this project is to implement a variant of SGD that will integrate with the current big data ecosystem. One recent advance in delay-tolerant SGD seems well suited for implementation on SAMOA [4].

[1] http://www.csie.ntu.edu.tw/~cjlin/papers/libmf.pd
[2] http://hazy.cs.wisc.edu/hazy/victor/Hogwild/
[3] http://hunch.net/~vw/
[4] http://static.googleusercontent.com/media/research.google.com/en//pubs/archive/43138.pdf

Adaptive + Boosting Vertical Hoeffding Tree

Hoeffding trees (aka,Very Fast Decision Trees) [1] are decision trees for streaming data. They can be used for classification of streams of unbounded data that needs to be analyzed very fast. The Vertical Hoeffding Tree is a particular implementation of this algorithm. The VHT is a parallel algorithm that works on distributed streaming environments.

The goal of this project is twofold. First, to extend the current implementation of the VHT to handle “concept drift”, that is, a change in the distribution of the attributes among the classes due to the evolution of the process generating the data [2].

Second, to implement a boosting version of the distributed algorithm [3]. Boosting is a meta-algorithm that trains an ensemble of classifiers. The challenge in parallelizing Boosting is that the models are dependent on each other in a linear chain (the output of the first model determines the input to the second model).

[1] http://homes.cs.washington.edu/~pedrod/papers/kdd00.pdf
[2] http://www.lsi.upc.edu/~abifet/R09-9.pdf
[3] http://en.wikipedia.org/wiki/Boosting_(machine_learning)

Regression Tree + GBDT

The regression tree is an algorithm for regression, often used to perform classification by simple thresholding. They are the basic building block of one of the most successful modern algorithms for classification, Gradient Boosted Decision Tree (GBDT). Several approaches to parallelization have been proposed (e.g., [2]).

The goal of this project is to implement a parallel version of GBDT in SAMOA.

[1] http://www-stat.stanford.edu/~jhf/ftp/trebst.pdf http://www-stat.stanford.edu/~jhf/ftp/stobst.pdf
[2] http://www.cslu.ogi.edu/~zak/cs506-pslc/sgradboostedtrees.pdf

Distributed Data Stream Mining using Coresets

Dealing with Big Data, the quantity of space needed to store it is very relevant. There are two main approaches: compression where we don’t lose anything, or sampling where we choose data that is more representative. Using compression, we may take more time and less space, so we can consider it as a transformation from time to space. Using sampling, we are losing information, but the gains in space may be in orders of magnitude.

In this project we will use coresets to reduce the complexity of Big Data problems. Coresets are small sets that provably approximate the original data for a given problem [1]. Using merge-reduce the small sets can then be used for solving hard machine learning problems in parallel.

[1] http://people.csail.mit.edu/dannyf/subspace.pdf

Distributed Data Stream Mining using Sketches

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. 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. An important challenge for data mining algorithm design is to make use of limited resources (time and memory).

In this project we plan to implement streaming structures called sketches to reduce the resources used in stream mining [1,2].

[1] http://blog.aggregateknowledge.com/2011/09/13/streaming-algorithms-and-sketches/
[2] http://github.com/addthis/stream-lib