DataSketches

An introduction

Claude N. Warren, Jr

January 12, 2020

Email: claude@xenei.com
Github: `https://github.com/Claudenw`
LinkedIn: `https://www.linkedin.com/in/claudewarren`
Research Gate: `https://www.researchgate.net/profile/Claude_Warren_Jr`

Overview

What is it

DataSketches [1] is an Apache Incubator [2] project. The project was started at Yahoo and accepted in the the Apache Incubator in March of 2019. It is currently in production use at several companies.

DataSketches is a high-performance library of stochastic streaming algorithms commonly called "sketches" in the data sciences. Sketches are small, stateful programs that process massive data as a stream and can provide approximate answers, with mathematical guarantees, to computationally difficult queries orders-of-magnitude faster than traditional, exact methods.

Apache DataSketches are available in Java, C++, and Python.

History

Sketching is a relatively recent development in the theoretical field of Stochastic Streaming Algorithms (Also known as “Approximate Query Processing”), which deals with algorithms that can extract information from a stream of data in a single pass (sometimes called “one-touch” processing) using various randomization techniques.

The term sketch, with its allusion to an artist’s sketch, has become the popular term to describe these algorithms and associated data structures that implement the theory.

Common Sketch Properties

- Single-pass, “one-touch” algorithms enable efficient processing in either real-time or batch.
- Mergeable algorithms enable parallel processing, which is critical for large systems.
- Space sub-linear algorithm not only start small but grow very slowly or not at all as the size of the input stream grows.
- Query results are approximate but within well defined error bounds that are user configurable by trading off sketch size with accuracy.
- Designed for Large-scale computing environments that must handle Big Data.(e.g., Hadoop, Pig, Hive, Druid, Spark).

Quality you expect from Apache

- Maven deployable and registered with The Central Repository.
- Comprehensive unit tests and testing tools are provided.
- Extensive documentation with the systems developer in mind.

Real-time Flurry, Before and After

Flurry: A system to manage data for mobile app developers. Captures data from apps and provides reporting to developers.

- Customers: $> 250K$ Mobile App Developers
- Data: 40-50 TB per day
- Platform: 2 clusters X 80 Nodes = 160 Nodes
- Node: 24 CPUs, 250GB RAM

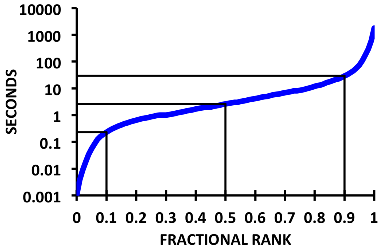
	Before Sketches	After Sketches
VCS* / Mo.	$\approx 80B$	$\approx 20B$
Result Freshness	Daily: 2 to 8 hours; Weekly: 3 days. Real-time Results Not Feasible!	15 seconds!

*Virtual Core Seconds

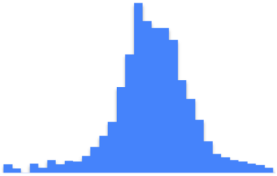
Quantiles

Estimates distribution, with well defined error bounds, of comparable values from a stream.

Enables queries of quantile(rank), rank(quantile) as well as normal or inverse Probability Mass Functions (PMF) or Cumulative Distribution Functions (CDF) of the distributions of any numeric value from your raw data in a single pass. This makes it relatively easy to produce frequency histograms.



$\xrightarrow{\textit{transform}}$



Java example: Using the quantile sketch

```
/* Create two sketches */
UpdateDoublesSketch time1Sketch =
   DoublesSketch.builder().build();
DoubleStream stream = // get a time boxed double
   stream
stream.forEach( time1Sketch::update )

UpdateDoublesSketch time2Sketch =
   DoublesSketch.builder().build();
DoubleStream stream = // get a time boxed double
   stream
stream.forEach( time2Sketch::update )
```

Java example: Merging sketches

```
/* Load stored sketches */
DoublesSketch timeSketch1 = ...
DoublesSketch timeSketch2 = ...

/* Merge them and report */
DoublesUnion union = DoublesUnion.builder().build();
union.update(time1Sketch);
union.update(time2Sketch);
DoublesSketch result = union.getResult();
```

Java example: Printing results

```java
/* print Min, Median, Max values */
System.out.println(Arrays.toString(result.getQuantiles(
    new double[] {0, 0.5, 1})));

/* define bins: (-inf ,-2), [-2,0), [0,2), [2,+inf) */
double[] bins = new double[] {-2, 0, 2};

/* Probability Histogram:  */
System.out.println(Arrays.toString(result.getPMF(bins)))

/* Freqency Histogram */
double[] histogram = result.getPMF(bins);
for (int i = 0; i < histogram.length; i++) {
     /* scale the fractions */
     histogram[i] *= result.getN();
  }
System.out.println(Arrays.toString(histogram));
```

Count Distinct / Count Unique

Solves Computational Challenges Associated with Unique Identifiers

- Estimating cardinality of a stream with many duplicates
- Performing set operations (e.g., Union, Intersection, and Difference) on sets of unique identifiers.
- Estimates of the error bounds of the result can be obtained directly from the result sketch

There are three families of Count Unique algorithms:

- The Theta Sketch Framework algorithms that are tuned for operation on the java heap or off-heap. Provides union, intersection and difference operations on sketches.
- The Hyper-Log Log (HLL) algorithms when sketch size is of utmost concern but can only perform union operations on sketches.
- The Compressed Probabilistic Counting (CPC) sketch. The CPC sketch beats the HLL sketch in terms of accuracy per stored space.

Frequent Items

Get the most frequent items (AKA heavy hitters) from a stream of items.

The accuracy of a Frequent Items Sketch is proportional to the configured size of the sketch, the larger the sketch, the smaller is the epsilon threshold that can detect Heavy Hitters.

Tuple Sketch

An Associative sketch that extends Theta Sketches to perform associative analysis.

Associative sketches that are useful for performing approximate join operations and extracting other kinds of behavior associated with unique identifiers by tracking additional summary data like impression counts or clicks on unique identifiers.

Sampling Sketches

A uniform sampling of a stream into a fixed size space.

This is a sketch that implements the famous Reservoir sampling algorithm. It supports mergability and uses Java Generics. VarOpt sampling extends the family to weighted sampling, additionally providing subset sum estimates from the sample with provably optimal variance.

Frequent Directions

Distributed, mergeable Singular Value Decomposition.

Part of a new separate sketches-vector package, Frequent Directions is in many ways a generalization of the Frequent Items sketch to handle vector data. This sketch computes an approximate singular value decomposition (SVD) of a matrix, providing a projection matrix that can be used for dimensionality reduction. SVD is a key technique in many recommender systems, providing shopping suggestions based on a customer's past purchases compared with other similar customers.

Additional Information

- DataSketches overview from Apache:
  `http://datasketches.apache.org/docs/TheChallenge.html`
- Downloads and Maven access:
  `http://datasketches.apache.org/docs/downloads.html`
- Source from GitHub: `https://github.com/apache?utf8=%E2%9C%93&q=datasketches&type=&language=`
- Research background and references:
  `http://datasketches.apache.org/docs/Research.html`
- Online discussion:
  `https://groups.google.com/forum/#!forum/sketches-user`
- Mailing lists can be found on the Apache incubator site for DataSketches:
  `https://incubator.apache.org/projects/datasketches.html`

Acknowledgements

I wish to thank Lee Rhodes, Distinguished Architect, Yahoo/Virgin Media for making the time to meet with me and explain how sketches work and for answering email questions about the same and also the DataSketches team:

- Alex Saydakov, Systems Developer, Yahoo/Verizon Media, joined 2015
- Jon Malkin, Ph.D., Scientist, Developer, Yahoo/Verizon Media, joined 2016
- Edo Liberty, Ph.D., Founder, HyperCube Technologies. Joined 2015
- Justin Thaler, Ph.D., Assistant Professor, Georgetown University, Computer Science. Joined 2015.
- Roman Leventov, Systems Developer for Druid, Metamarkets, joined 2018
- Graham Cormode, Ph.D., Professor, University of Warwick, Computer Science, joined 2017
- Jelani Nelson, Ph.D., Professor, U.C. Berkeley, joined 2019
- Daniel Ting, Ph.D., Sr Scientist, Salesforce, joined 2019
- Christopher Musco, Ph.D., Assistant Professor NYU, Theoretical Computer Science, Lin Algebra. Joined 2018

References I

[1] The Apache Foundation. *Apache DataSketches: A software library of stochastic streaming algorithms*. 2019. URL: http://datasketches.apache.org.

[2] The Apache Foundation. *Datasketches Project Incubation Status*. URL: https://incubator.apache.org/projects/datasketches.html.