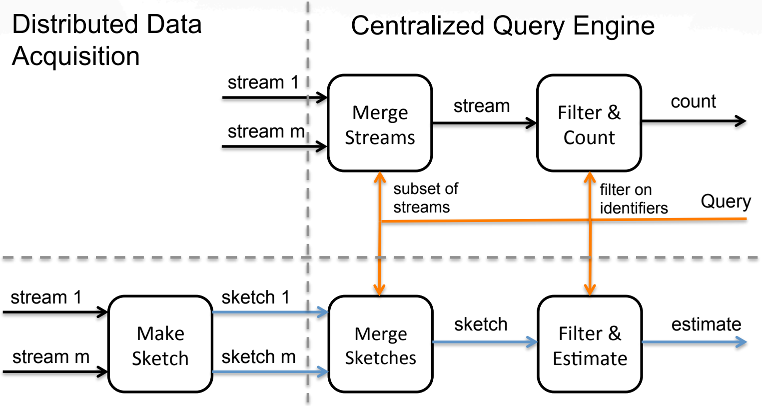
A Framework for Estimating Stream Expression Cardinalities

Anirban Dasgupta, Kevin Lang, Lee Rhodes, Justin Thaler

Motivating Task

- Distributed data acquisition.
- Count-Distinct queries with predicates.
- Example: how many unique IP addresses accessed servers in either UK and France yesterday, not counting those on the spam-bot list that we just got.

Brute Force Solution Contrasted with Sketches



Remarks

- HyperLogLog sketches don't work because of the late-arriving predicate.
- K'th Minimum Value sketches [Beyer *et al*] would work in principle.
  - KMV sketch is the set of k+1 smallest hashed ID's.
  - KMV estimate is k/m_{k+1}.
- However, practical difficulties arise in large real-world organizations.

Difficulty 1: Non-matching values of k.

[k determines # of samples in sketch.]

- Team A: k = 10,000
- Team B: k = 2,000 (this year)
- Team B: k = 1,000 (last year)

Question: how to produce estimates spanning both teams and both years? Note: reducing all k's to 1,000 means throwing away data.

Difficulty 2: Alternate Sketching Algorithms

- Team C: used Adaptive Sampling because higher throughput.
- Team D: used a novel algorithm that downsamples short as well as long streams [see pKMV in paper].

Now there are three different kinds of sketches. Are they mergeable? Who knows, but the boss say we need to compute company-wide estimates.

Difficulty 3: Exotic Sketching Algorithms

- Team E: used a novel algorithm that has even better latency and throughput than Adaptive Sampling [see Alpha Algorithm in paper].
- Team F: used a plausible-sounding modification of KMV called $QS\backslash C$.

These algorithms are sensitive to input order, hence difficult to analyze even for single streams, much less unions. Now how does one compute company-wide estimates?

Our Theoretical Contribution

- Identified ”real” reason why KMV estimates are unbiased:
- A simple property called “1-goodness”.
- Also 1-good: pKMV, Adaptive Sampling, Alpha Algorithm, and many others.
- Brings them into a common mathematical framework.
- More importantly, allows a real-world system to freely intermingle and MERGE all of these sketch types.

Theta Sketch Framework

- Parameterized by a "Threshold Choosing Function" T() that maps streams to thresholds in $(0, 1]$.
- $T() = m_{k+1}$ instantiates the framework as KMV.
- A different T() instantiates it as Adaptive Sampling.
- Other choices of T() result in novel schemes such as pKMV and Alpha Algorithm.

Theta Sketch Framework, cont'd

- T() maps streams to thresholds in $(0, 1]$.
- Let A be a stream [already hashed to values in $[0, 1)$].
- Sketch is a pair (θ, S), where

$$\theta = T(A).$$
$$S = \{h \in A \text{ s.t. } h < \theta\}.$$

- The cardinality estimate is $\hat{n} = |S|/\theta$.

Theta Sketch Framework, cont'd again

- $\hat{n} = |S|/\theta = \sum_n S_i/T()$, where S_i is Bernoulli indicator variable for membership in the set S.
- Therefore $E(\hat{n}) = \sum_n E(S_i/T())$,
- so $E(S_i/T())$ = 1 would imply $\hat{n} = n$, [in other words, the cardinality estimate would be unbiased.]

Fixed-Threshold Sampling

- Horvitz-Thompson: if $T()$ were a constant F, then $E(S_i/T()) = F/F = 1$.
- Unfortunately, fixed-threshold sampling uses $O(N)$ space and is therefore not sketch-like.

Better functions $T()$ depend in complicated ways on the hash values in the stream.

Our Main Theorem

- Theorem: if every "fix-all-but-one projection" of T() is "1-good", then $E(S_i/T()) = 1$ so $E(\hat{n}) = n$.
- In other words, the sketches provide unbiased estimates.
- Coming up: definitions of the quoted terms.

Definition: fix-all-but-one projection of T(stream)

- Pick any label L in the stream.
- Freeze the hash values of all other labels.
- Consider the univariate function T(x) where x is the hash value of L.

Formal Definition of 1-goodness

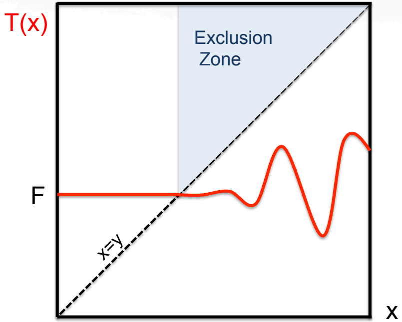
A univariate function T(x) is 1-good iff there exists a constant F such that:

if x < F then T(x) = F

else T(x) <= x.

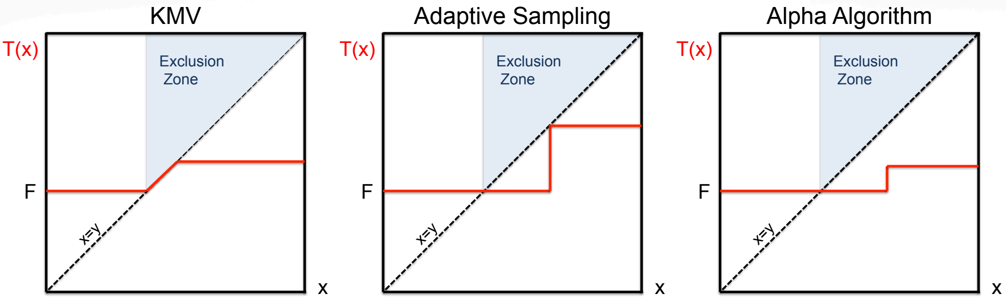
That definition is VERY dry, so ...

Diagram Illustrating 1-goodness



Vary x from 0 to 1. If $T(x)$ is constant until the $x = y$ line, and then avoids the "exclusion zone", then $T(x)$ is 1-good.

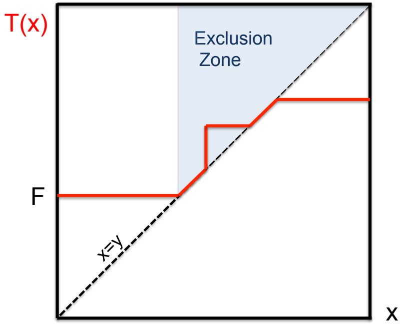
Why Three Sketching Algorithms are Unbiased



Previously, each had a separate proof involving lots of algorithmic-specific math.

Contrapositive of Main Theorem

Any T() that yields biased estimates possesses at least one fix-all-but-one projection that is not 1-good.



Sketch Merging

- Framework uses a special *minimum-of-θ's* merging rule.
- $\theta_u = \min_m \theta_j$.
- $S_u = \{h \in \cup_m S_j \text{ s.t. } h < \theta_u\}$.
- Theorem: subject to several conditions [see paper], the estimation error of m-o-θ's union sketches is at least as small as sketches created directly from concatenated input streams.

Main Result about Sketch Merging

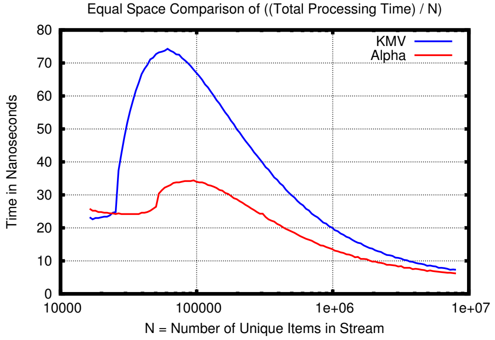
- Theorem: If 1-good sketches are combined using the minimum of θ's rule, then the result is a 1-good sketch of the union of the input streams.
- Corollary: m-o-θ's union sketches provide unbiased estimates.
- Holds even for non-matching k's and multiple base algorithms.
- The big-organization problem is solved!

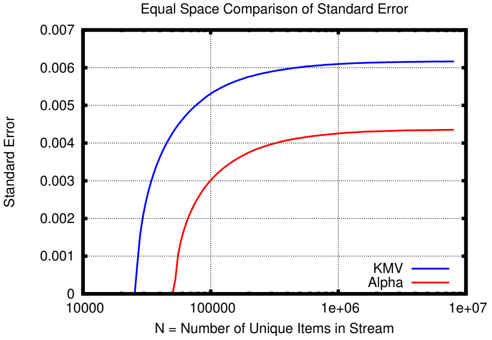
Novel Base Algorithm: pKMV

- Some real-world systems contain an astronomical number of very short streams.
- Problem: When $|\text{stream}| < k$, KMV saves no space.
- Solution: pKMV, for which $T() = \min(m_{k+1}, p)$.
- If e.g. $p = 1/8$, would save at least factor of 8 space, and more than that for long streams.

Novel Base Algorithm: Alpha

The “Alpha Algorithm” is 1-good and provides KMV-like behavior without a heap data structure or QuickSelect. Results of Equal-Space Comparison:





YAHOO!

Summary and Conclusion

- 1-goodness is a very simple sufficient condition for unbiasedness of KMV-like sketching algorithms.
- Our theoretical results permit different kinds of sketches to co-exist and be combined within a single real-world Big Data system.
- Systems of this type have been built by Yahoo.
- The sketching code that lies at the heart of these systems is available as an open-source library at `http://datasketches.github.io`.