Noel Welsh

_.underscore

Making Big Data Small

Strata Conference London 2012

Goal: Introduce a relatively unknown class of algorithms for handling big data

Big Data is easy

Just use Hadoop

Hadoop is complex, slow, and batch

We want simple, fast, and real-time

Answer: Streaming Algorithms

Key feature: process data one item at a time

Can apply to real-time or batch processing

Summarise	Classify	Use
Set cardinality	Perceptron	Bandit algorithms
Frequent items	Linear SVM	Reinforcement learning
Quantiles	SGD	
Clustering	SVI	
etc.	etc.	

Analytics

Summarise

Set cardinality
Frequent items
Quantiles
Clustering
etc.

Machine Learning

Classify

Perceptron
Linear SVM
SGD
SVI
etc.

Use

Bandit algorithms
Reinforcement learning

Machine Learning

Use

Bandit algorithms

Analytics

Summarise

Distinct values
Frequent items
Quantiles
Clustering
etc.

Analytics

Summarise

Distinct values

Overview

- Hash Functions
- The Distinct Values Problem
- *k*-Minimum Values

Hash Functions

Streaming algorithms
love hash functions!
Let's do a quick review

Deterministic

Uniform distribution

Bit values are independent

In Practice?

Use Murmur Hash 3

Distinct Values

Distinct Values

- Count the size of a set. "How many users arrived from the Strata website?"

The Joy of Sets

- With set algebra we can answer many questions of interest
- How many users came from Strata OR Facebook?
- How many users came from Facebook AND purchased?

An Analytics Platform

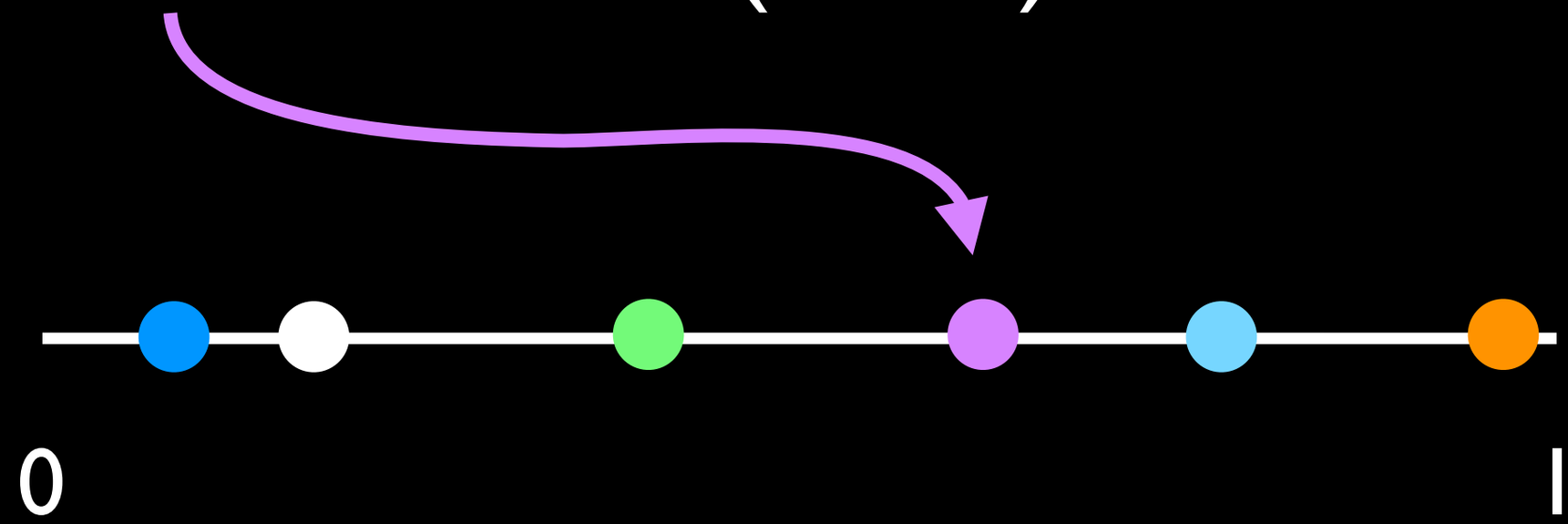
- We can build a fairly general analytics platform with just distinct values and set operations!

Many Roads

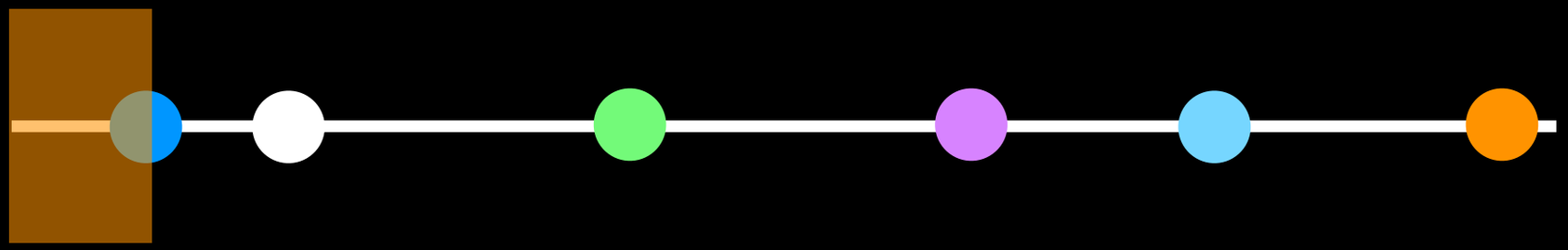
- A lot of research has been done
- Flajolet-Martin sketches (LogLog and HyperLogLog) are popular
- Optimal (but complex) algorithm published in 2010

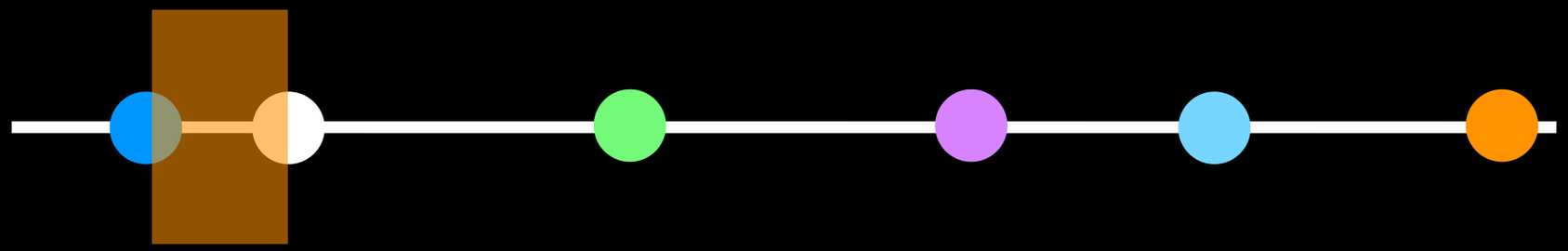
k-Minimum Values

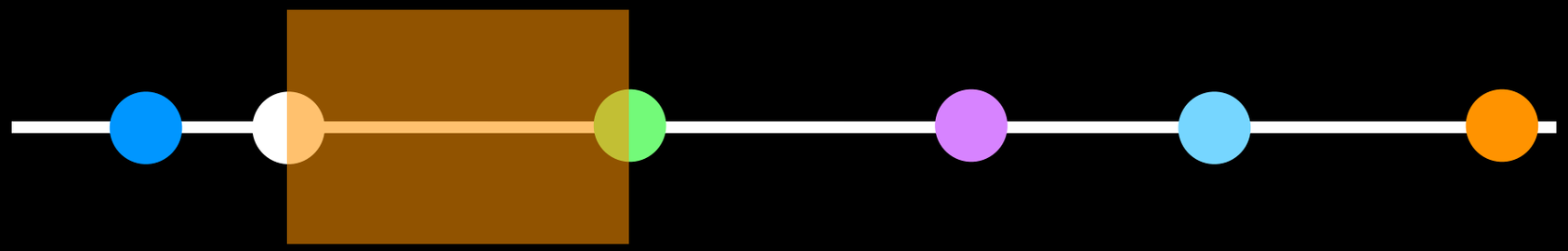
index = hash(data) / maxHash

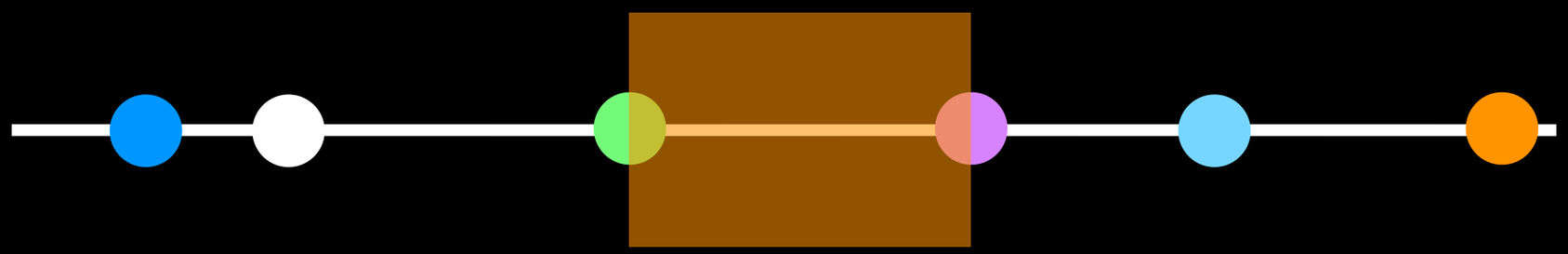


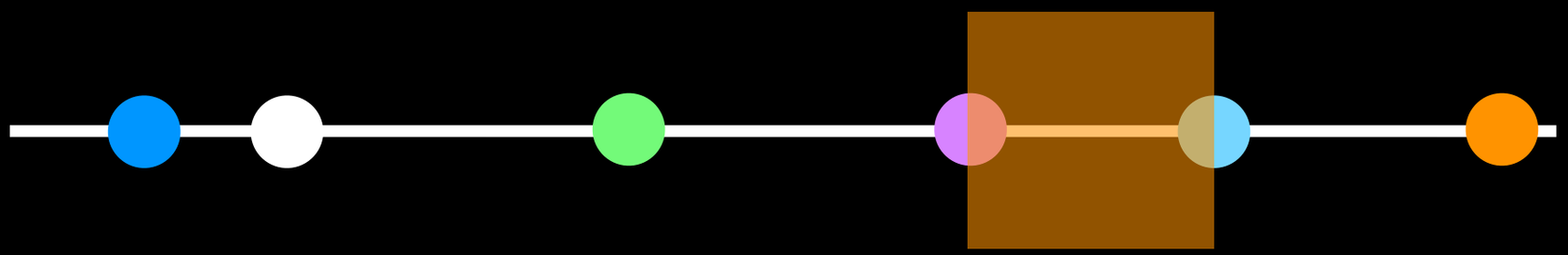
Average distance between elements inversely proportional to cardinality

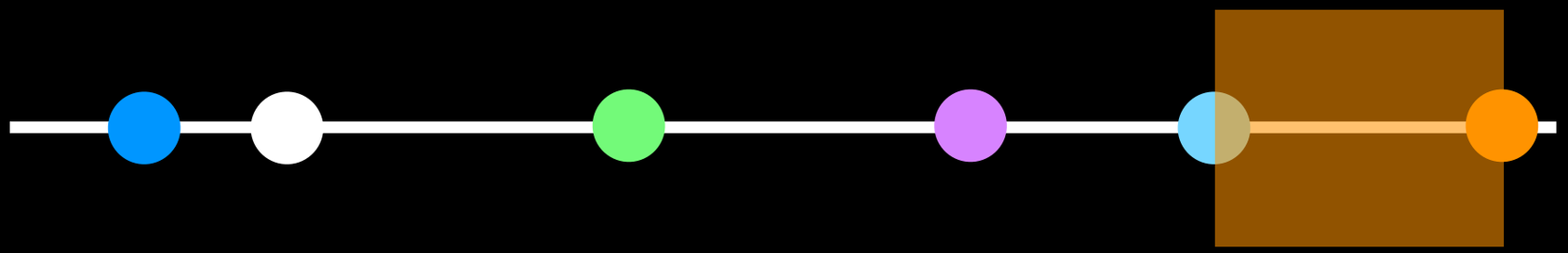


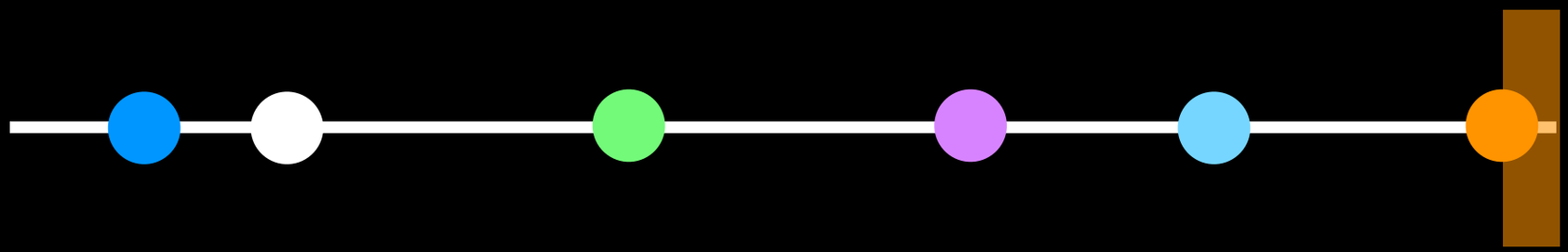










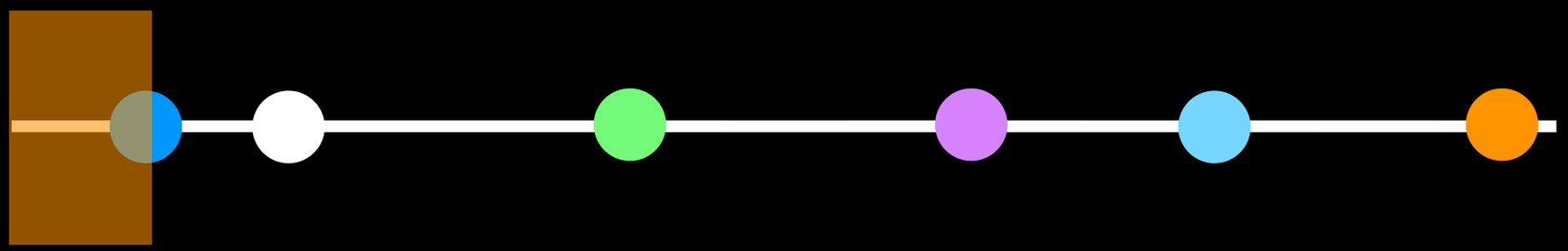


Can’t store all these distances

Big Idea

- Store minimum value. This gives us *one* distance
- Estimate size of set

$$|S| = \frac{1}{\text{minimum}}$$

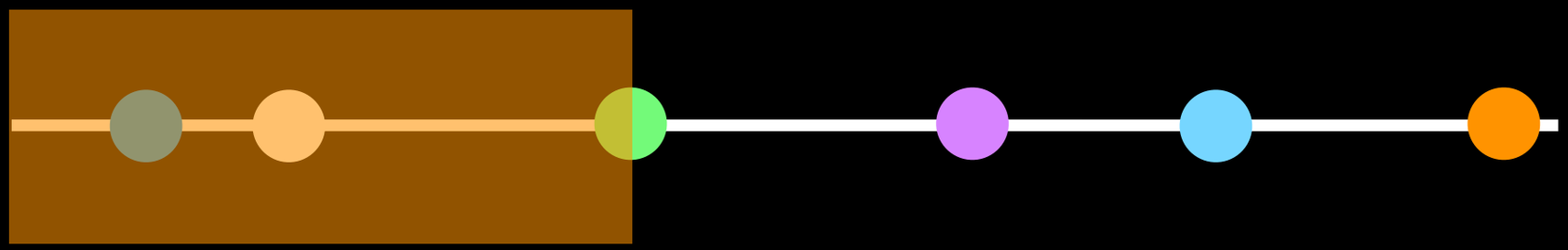


Very noisy!

Refinement

- Store *k* minimum values
- Estimate cardinality as

$$|S| = \frac{k-1}{\text{largest value stored}}$$



$k = 3$

Error Rate

$$\mathbb{E}\left[\frac{|S|_{est} - |S|}{|S|}\right] \approx \sqrt{\frac{2}{\pi(k-2)}}$$

- Independent of size of set
- See papers for other error bounds

Example

- Storing $k = 1024$ values (typically 4K) gives expected error of 2.5%

Set Algebra

- Set union is just the *k* minimum values from the union of the two sets
- Set intersection from Jaccard coefficient
- Set difference if we add counters to each element we store

Summary

- *k*-Minimum Values is simple
- Space usage and error are small
- Real-time processing extremely feasible
- Look at (Hyper)LogLog if applying for real

More

Other Algorithms

- We've only touched the surface
- Quantiles, clustering, graph properties, etc.
- Online learning is an area I'm excited about. Goes beyond summarising data to taking actions.

Code

- Clearspring's stream-lib implements the most common streaming analytics algorithms in Java.
https://github.com/clearspring/stream-lib

Writing

- Lots of blog posts, tutorials, etc. Ask Google
- Alex Smola's course is a good overview http://alex.smola.org/teaching/berkeley2012/streams.html
- *k*-Minimum Values is in http://www.mpi-inf.mpg.de/~rgemulla/publications/beyer07distinct.pdf

Talking

- Interested? Let's chat!

Me

- Slides will be on noelwelsh.com
- noel@underscoreconsulting.com
- @noelwelsh