Data Mining and Analysis

Streaming Data

CSCE 676 :: Fall 2019
Texas A&M University
Department of Computer Science & Engineering
Prof. James Caverlee

Resources

MMDS Chapter 4 + slides

- http://infolab.stanford.edu/~ullman/mmds/ch4.pdf
- http://www.mmds.org/mmds/v2.1/ch04-streams1.pdf
- http://www.mmds.org/mmds/v2.1/ch04-streams2.pdf

Carlos Castillo course on Data Mining [https://github.com/chatox/data-mining-course]

What is a data stream?

A potentially infinite sequence of data points

Examples:

- Web click-stream data
- Stock quotes
- Sensor data (e.g., temperature, air pressure)
- Network monitoring data
- ...

Key Properties

Unbounded size

Data cannot be persisted on disk

Only summaries can be stored

Transient

Single pass over the data

Sometimes real-time processing is needed

Dynamic

May require incremental updates

May require forgetting old data

Concepts “drift”

Temporal order is often important

Applications

Mining query streams

A search engine wants to know what queries are more frequent today than yesterday

Mining click streams

Amazon wants to know when one of its pages starts getting an unusual number of hits per hour

Mining social network news feeds

Twitter or Facebook wants to show trending topics

Applications

Sensor networks

- Many sensors feeding into a central controller

Telephone call records

- Data feeds into customer bills as well as settlements between telephone companies

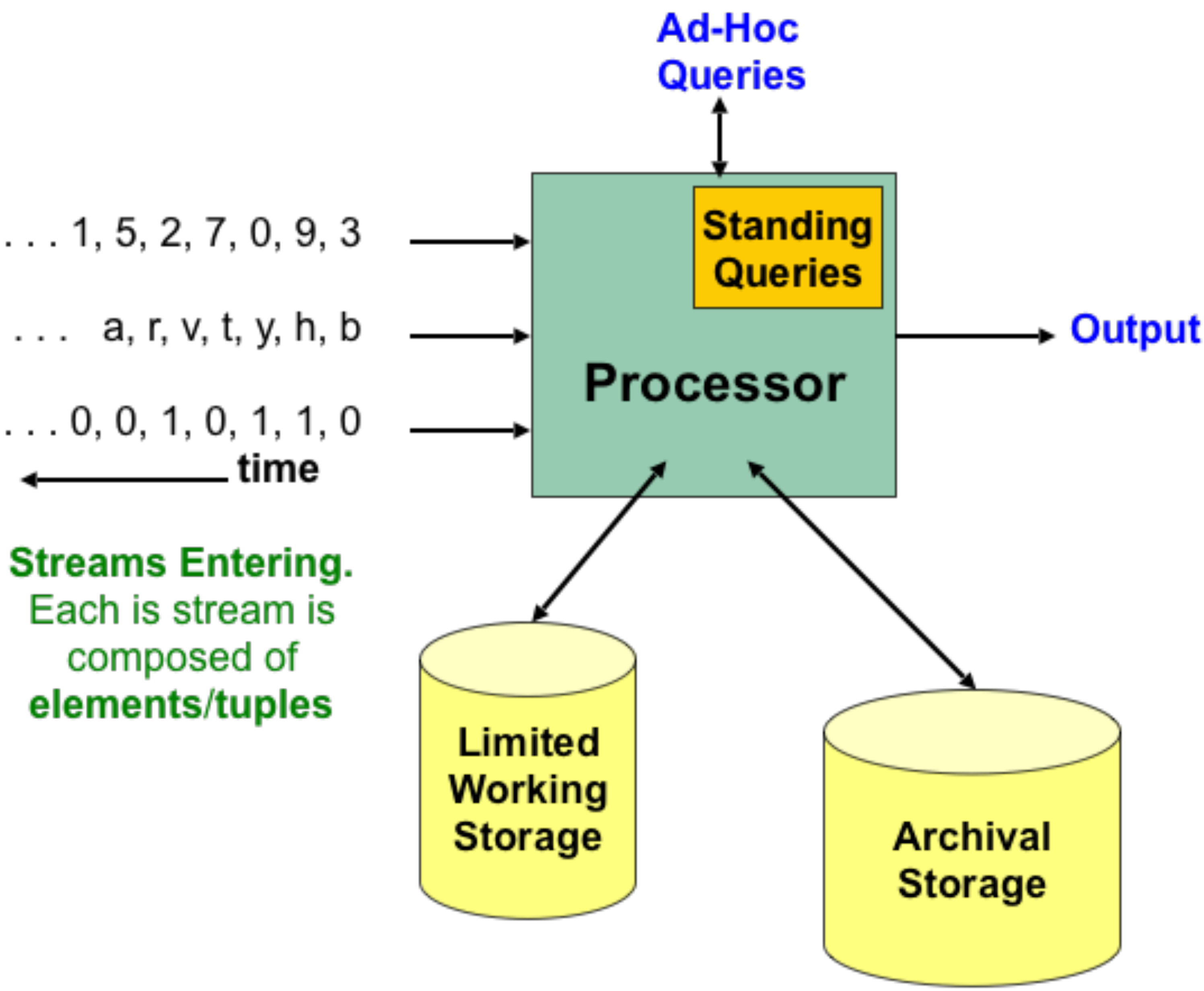
IP packets monitored at a switch

- Gather information for optimal routing
- Detect denial-of-service attacks

Why do we need new algorithms?

	Traditional	Stream
passes	multiple	single
processing time	unlimited	restricted
memory	disk	main memory
results	typically accurate	approximate
distributed	typically not	often

General Stream Processing Model



Problems on Data Streams?

Sampling data from a stream

Queries over sliding windows

Filtering a data stream

Counting distinct elements

Finding frequent elements

…

Filtering Data Streams

Each element of data stream is a tuple

Given a list of keys S

Determine which tuples of stream are in S?

Bloom filters!

Sampling a fixed proportion

Sampling a fixed proportion

Example stream: `<user, query, timestamp>` from a search engine query log

Suppose we have space to store 1/r of the stream

- E.g., 1/10th, 1/100th, 1/1000th, ...

Naive solution:

- Generate uniform random number in 0 ... (r-1)
- If the number is 0, keep the item

What can we do with this sample?

- Estimate the most frequent query
  - Pick the most frequent in the sample
- Estimate the frequency of a query
  - Multiply the observed frequency by r
- Do people ask query q?
  - Approximate answer (with some error)

Question Time

We want to tell if we have seen item q

Suppose we have seen n items so far

We have sampled a fraction 1/r

Suppose item q appears with prob p(q)

What is the probability of:

- False positive? (item q **was not** in the stream but we said it **was**) ZERO
- False negative? (item q **was** in the stream but we said it **was not**) $(1-p(q))^{n/r}$

But there are questions we **cannot** answer with the naive approach

Example: What fraction of queries by an average search engine user are duplicates?

Suppose each user issues

x queries once and d queries twice

In total: **x + 2d queries**

Correct answer = **d/(x+d)**

Proposed solution: We keep 1/10th of the queries

Sample will contain x/10 of the singleton queries at least once

Sample will contain 2d/10 of the duplicate queries at least once

Sample will contain **d/100 pairs of duplicates**

$$d/100 = 1/10 * 1/10 * d$$

Of the d duplicates, 18d/100 will be seen once

$$18d/100 = ((1/10 * 9/10) + (9/10 * 1/10)) * d$$

So the sample answer is:

Observed duplicates

$$\frac{\frac{d}{100}}{\frac{x}{10} + \frac{18d}{100} + \frac{d}{100}} = \frac{d}{10x + 19d}$$

Observed singletons

Observed duplicates

WRONG!

Solution: Sample users!

Pick 1/10th of users and take all their searches in the sample

How?

Hashing

Given `<user, query, timestamp>`

Compute h(user) —> 0, 1, … (r-1)

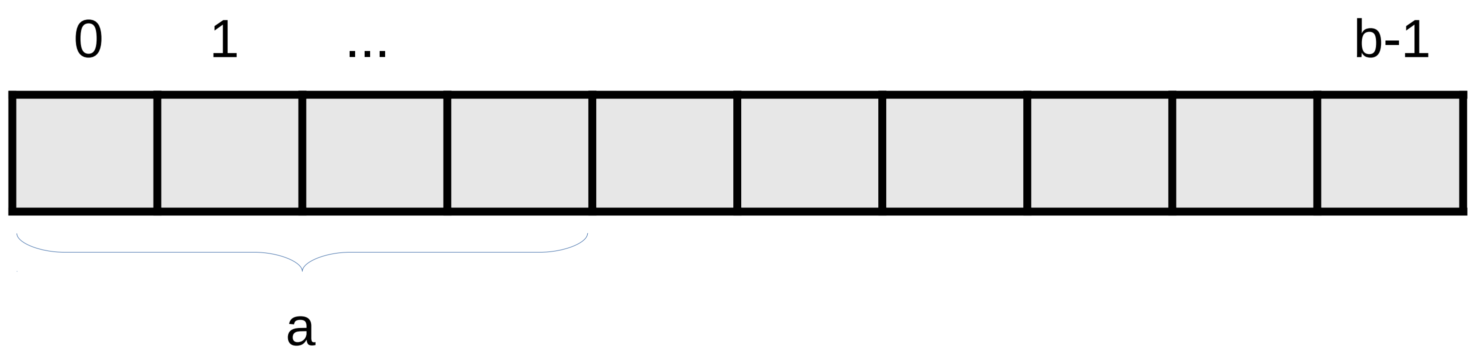
Keep tuple if hash value is 0

In general

To sample a/b of a stream by key

Compute hash(key) —> 0, 1, (b-1)

Keep if h(key) < a



Sampling a fixed-size sample

A fixed size sample

We normally do not know the stream size

We just know how much storage space we have

Suppose we have storage space s and want to maintain a random sample S of size $s=|S|$

Requirement: after seeing n items, each of the n items should be in our sample with probability s/n

- No item should have an advantage or disadvantage

Bad solutions

Suppose stream = <a, f, e, b, g, r, u, …>

Requirement: after seeing n items, each of the n items should be in our sample with probability s/n

Suppose s=2

> Always keep first two? No, because then p(a) = 1, while p(e) = 0
>
> Always keep last two? No, because then p(a) = 0, while p(u) = 1

Sample some? But which? Then evict some? But which?

Reservoir sampling

Elements $x_1, x_2, x_3, \ldots, x_i, \ldots$

1. Store all first s elements $x_1, x_2, \ldots, x_s$
2. Suppose element x_n arrives

   With probability 1-s/n, ignore this element

   With probability s/n:

   - Discard a random element from the reservoir
   - Insert element x_n into the reservoir

Example

Input is <a, b, c, …>

Suppose s=2

We have just processed element 3 = “c”

What is:

Probability “a” is in the sample?

Probability “b” is in the sample?

Probability “c” is in the sample?

Proof by induction

Inductive hypothesis: after n elements seen, each of them is sampled with probability s/n

Base case: after we see n=s elements, the sample S has the desired property

Each out of n=s elements is in the sample with probability s/s = 1

Proof by induction

Inductive hypothesis: after n elements seen, each of them is sampled with probability s/n

Inductive step: element x_{n+1} arrives

What is the probability that an already-sampled element x_i stays in the sample?

$$\underbrace{\left(1 - \frac{s}{n+1}\right)}_{x_{n+1} \text{ not sampled}} + \underbrace{\left(\frac{s}{n+1}\right)}_{x_{n+1} \text{ sampled}} \cdot \underbrace{\left(\frac{s-1}{s}\right)}_{x_i \text{ not evicted}} = \frac{n}{n+1}$$

Proof by induction

Tuple x_{n+1} is sampled with probability s/(n+1)

Tuples x_i with i<=n

Were in the sample with probability s/n

Stay in the sample with probability n/(n+1)

Hence, in the sample with probability

$$\frac{s}{n} \cdot \frac{n}{n+1} = \frac{s}{n+1}$$