Summarizing and mining inverse distributions on data streams via dynamic inverse sampling

Presented by

Graham Cormode

cormode@bell-labs.com

S. Muthukrishnan

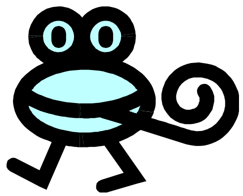
muthu@cs.rutgers.edu

Irina Rozenbaum

rozenbau@paul.rutgers.edu

Outline

- Defining and motivating the Inverse Distribution
- Queries and challenges on the Inverse Distribution
- Dynamic Inverse Sampling to draw sample from Inverse Distribution
- Experimental Study



Data Streams & DSMSs

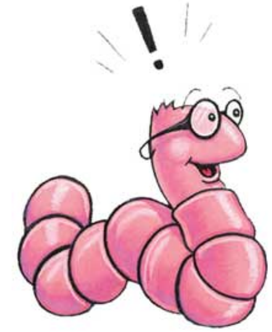
- Numerous real world applications generate data streams:
  - IP network monitoring
  - click streams
  - Telecommunications
  - financial transactions
  - sensor networks
  - text streams at application level, etc.

- Data streams are characterized by massive data volumes of transactions and measurements at high speeds.

- Query processing is difficult on data streams:
  - We cannot store everything, and must process at line speed.
  - Exact answers to many questions are impossible without storing everything
  - We must use approximation and randomization with strong guarantees.

- Data Stream Management Systems (DSMS) summarize streams in small space (samples and sketches).

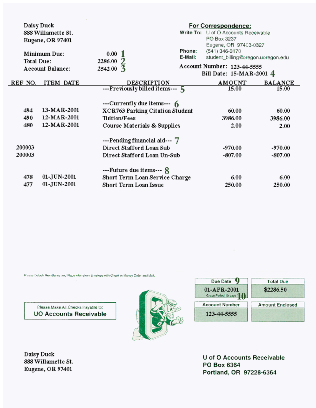
DSMS Application:
IP Network Monitoring



- Needed for:
  - network traffic patterns identification
  - intrusion detection
  - reports generation, etc.

- IP traffic stream:
  - Massive data volumes of transactions and measurements:
    - over 50 billion flows/day in AT&T backbone.
  - Records arrive at a fast rate:
    - DDoS attacks - up to 600,000 packets/sec

- Query examples:
  - heavy hitters
  - change detection
  - quantiles
  - Histogram summaries



Daisy Duck
888 Willamette St.
Eugene, OR 97401

Minimum Due: 0.00 1
Total Due: 2286.00 2
Account Balance: 2542.00 3

For Correspondence:
Write To: U of O Accounts Receivable
PO Box 3237
Eugene, OR 97403-0327
Phone: (541) 346-3170
E-Mail: student_billing@oregon.uoregon.edu

Account Number: 123-44-5555
Bill Date: 15-MAR-2001 4

REF NO.	ITEM DATE	DESCRIPTION	AMOUNT	BALANCE
		---Previously billed items--- 5	15.00	15.00
		---Currently due items--- 6		
494	13-MAR-2001	XCR763 Parking Citation Student	60.00	60.00
490	12-MAR-2001	Tuition/Fees	3986.00	3986.00
480	12-MAR-2001	Course Materials & Supplies	2.00	2.00
		---Pending financial aid--- 7		
200003		Direct Stafford Loan Sub	-970.00	-970.00
200003		Direct Stafford Loan Un-Sub	-807.00	-807.00
		---Future due items--- 8		
478	01-JUN-2001	Short Term Loan Service Charge	6.00	6.00
477	01-JUN-2001	Short Term Loan Issue	250.00	250.00

Please Detach Remittance and Place into return Envelope with Check or Money Order and Mail.

Please Make All Checks Payable to:
UO Accounts Receivable

Due Date 9	Total Due
01-APR-2001 Grace Period 10 days 10	$2286.50
Account Number	Amount Enclosed
123-44-5555	

Daisy Duck
888 Willamette St.
Eugene, OR 97401

U of O Accounts Receivable
PO Box 6364
Portland, OR 97228-6364

Forward and Inverse Views

Consider the IP traffic on a link as packet p representing (i_p, s_p) pairs where i_p is a source IP address and s_p is a size of the packet.

Problem A.

Which IP address sent the most bytes?

That is , find i such that $\sum_{p|i_p=i} s_p$ is maximum.

Forward distribution.

Problem B.

What is the most common volume of traffic sent by an IP address?

That is , find traffic volume W s.t $|\{i|W = \sum_{p|i_p=i} s_p\}|$ is maximum.

Inverse distribution.

The Inverse Distribution

If f is a discrete distribution over a large set X, then inverse distribution, $f^{-1}(i)$, gives fraction of items from X with count i.

- Inverse distribution is $f^{-1}[0\ldots N]$,

$$f^{-1}(i) = \text{fraction of IP addresses which sent } i \text{ bytes.}$$
$$= |\{ x : f(x) = i, i^{-1} \neq 0\}|/|\{x : f(x)^{-1} \neq 0\}|$$

$$F^{-1}(i) = \text{cumulative distribution of } f^{-1}$$
$$= \sum_{j > i} f^{-1}(j) \quad [\text{sum of } f^{-1}(j) \textit{ above } i]$$

- Fraction of IP addresses which sent < 1KB of data $= 1 - F^{-1}(1024)$
- Most frequent number of bytes sent = i s.t. $f^{-1}(i)$ is greatest
- Median number of bytes sent = i s.t. $F^{-1}(i) = 0.5$

Queries on the Inverse Distribution

- Particular queries proposed in networking map onto f^{-1},
  - $f^{-1}(1)$ (number of flows consisting of a single packet) indicative of network abnormalities / attack [Levchenko, Paturi, Varghese 04]
  - Identify evolving attacks through shifts in Inverse Distribution [Geiger, Karamcheti, Kedem, Muthukrishnan 05]
- Better understand resource usage:
  - what is dbn. of customer traffic? How many customers < 1MB bandwidth / day? How many use 10 – 20MB per day?, etc. → Histograms/ quantiles on inverse distribution.
- Track most common usage patterns, for analysis / charging
  - requires heavy hitters on Inverse distribution
- Inverse distribution captures fundamental features of the distribution, has not been well-studied in data streaming.

Forward and Inverse Views on IP streams

Consider the IP traffic on a link as packet p representing (i_p, s_p) pairs where i_p is a source IP address and s_p is a size of the packet.

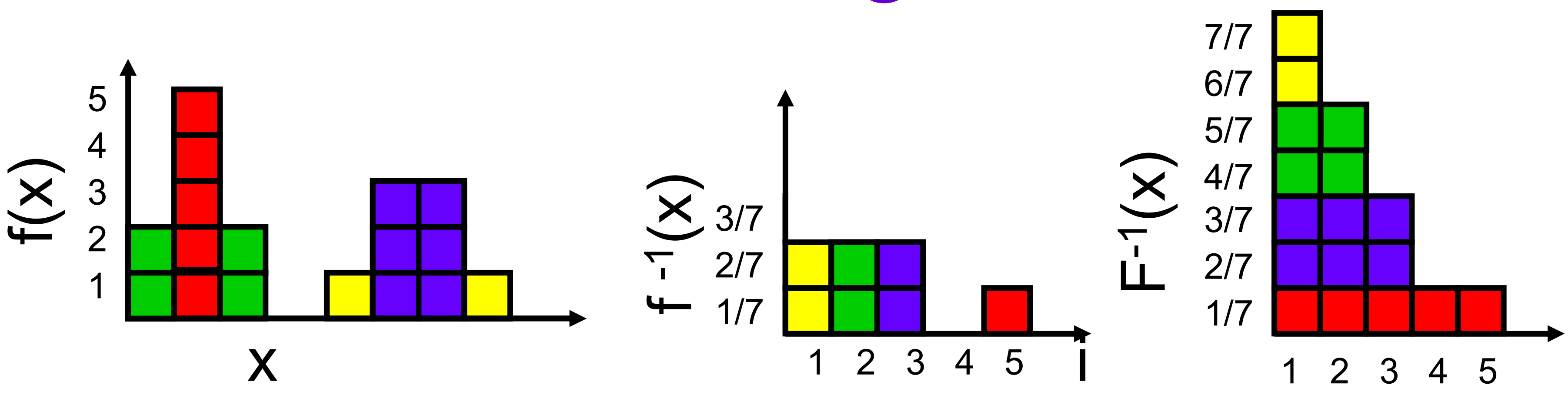
Forward distribution:

- Work on $f[0 \dots U]$ where $f(x)$ is the number of bytes sent by IP address x.
- Each new packet (i_p, s_p) results in $f[i_p] \leftarrow f[i_p] + s_p$.
- Problems:
  - $f(i) = ?$
  - which $f(i)$ is the largest?
  - quantiles of f ?

Inverse distribution:

- Work on $f^{-1}[0 \dots K]$
- Each new packet results in $f^{-1}[f[i_p]] \leftarrow f^{-1}[f[i_p]] - 1$ and $f^{-1}[f[i_p] + s_p] \leftarrow f^{-1}[f[i_p] + s_p] + 1$.
- Problems:
  - $f^{-1}(i) = ?$
  - which $f^{-1}(i)$ is the largest?
  - quantiles of f^{-1} ?

Inverse Distribution on Streams: Challenges I



- If we have full space, it is easy to go between forward and inverse distribution.
- But in small space it is much more difficult, and existing methods in small space don't apply.

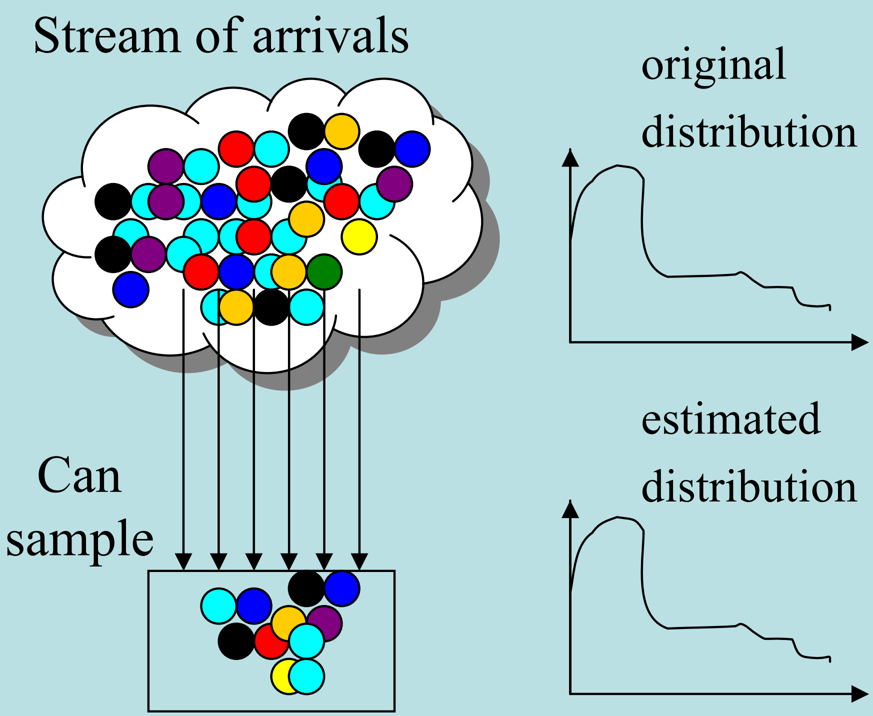
- Find f(192.168.1.1) in small space, with query give *a priori* – easy: just count how many times the address is seen.
- Find $f^{-1}(1024)$ – is provably hard (can't find exactly how many IP addresses sent 1KB of data without keeping full space).

Inverse Distribution on Streams: Challenges II, deletions

How to maintain summary in presence of insertions and deletions?

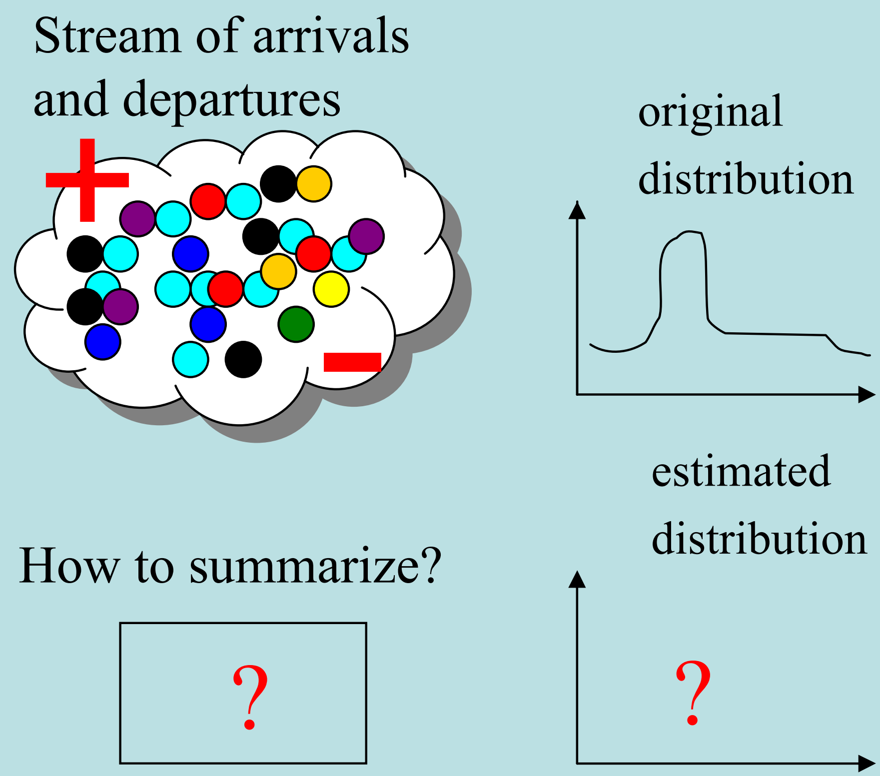
Insertions only

updates $s_p > 0$



Insertions and Deletions

updates s_p can be arbitrary

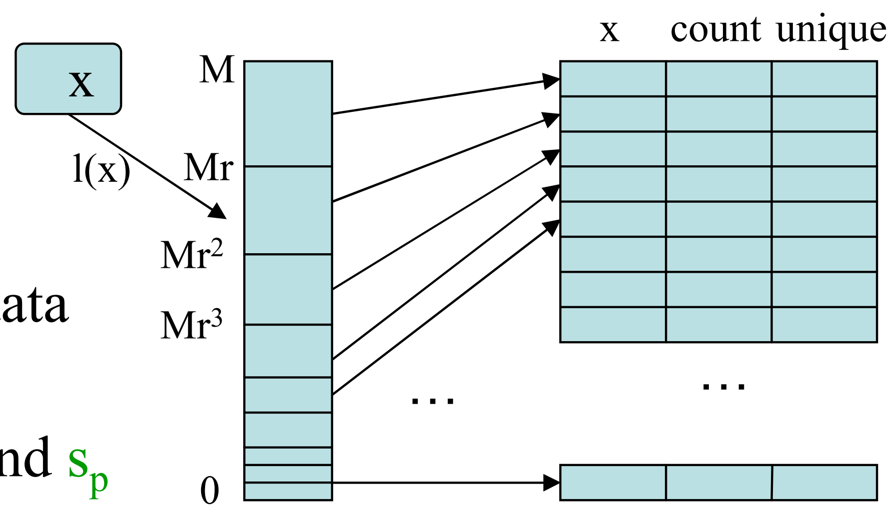


Our Approach: Dynamic Inverse Sampling

- Many queries on the forward distribution can be answered effectively by drawing a sample.
  - Draw an x so probability of picking x is $f(x) / \sum_y f(y)$
- Similarly, we want to draw a sample from the inverse distribution in the centralized setting.
  - draw (i,x) s.t. $f(x)=i, i\neq 0$ so probability of picking i is $f^{-1}(i) / \sum_j f^{-1}(j)$ and probability of picking x is uniform.
- Drawing from forward distribution is "easy": just uniformly decide to sample each new item (IP address, size) seen
- Drawing from inverse distribution is more difficult, since probability of drawing $(i,1)$ should be same as $(j,1024)$

Dynamic Inverse Sampling: Outline

- Data structure split into levels
- For each update (i_p, s_p):
  - compute hash $l(i_p)$ to a level in the data structure.
  - Update counts in level $l(i_p)$ with i_p and s_p

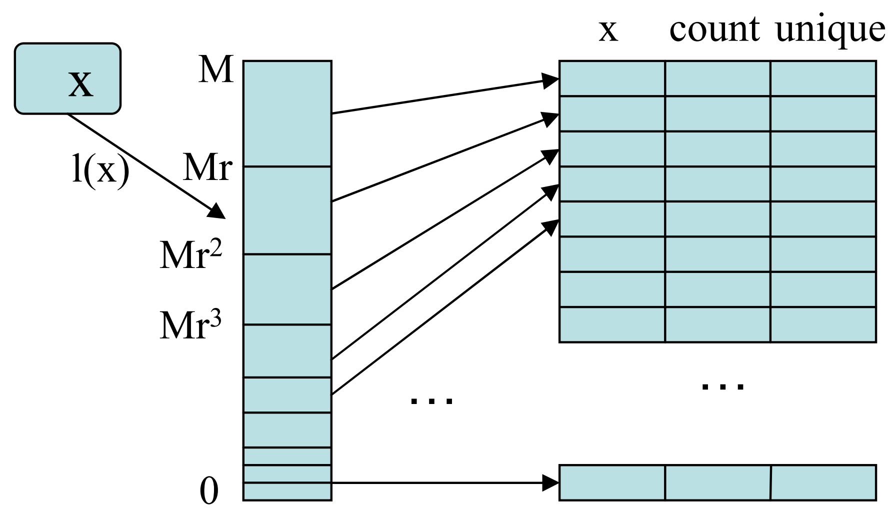


- At query time:
  - probe the data structure to return $(i_p, \Sigma s_p)$ where i_p is sampled uniformly from all items with non-zero count
  - Use the sample to answer the query on the inverse distribution.

Hashing Technique

Use hash function with exponentially decreasing distribution:
Let h be the hash function and r is an appropriate const < 1

$$\Pr[h(x) = 0] = (1-r)$$
$$\Pr[h(x) = 1] = r\,(1-r)$$
$$\ldots$$
$$\Pr[h(x) = l] = r^l(1-r)$$



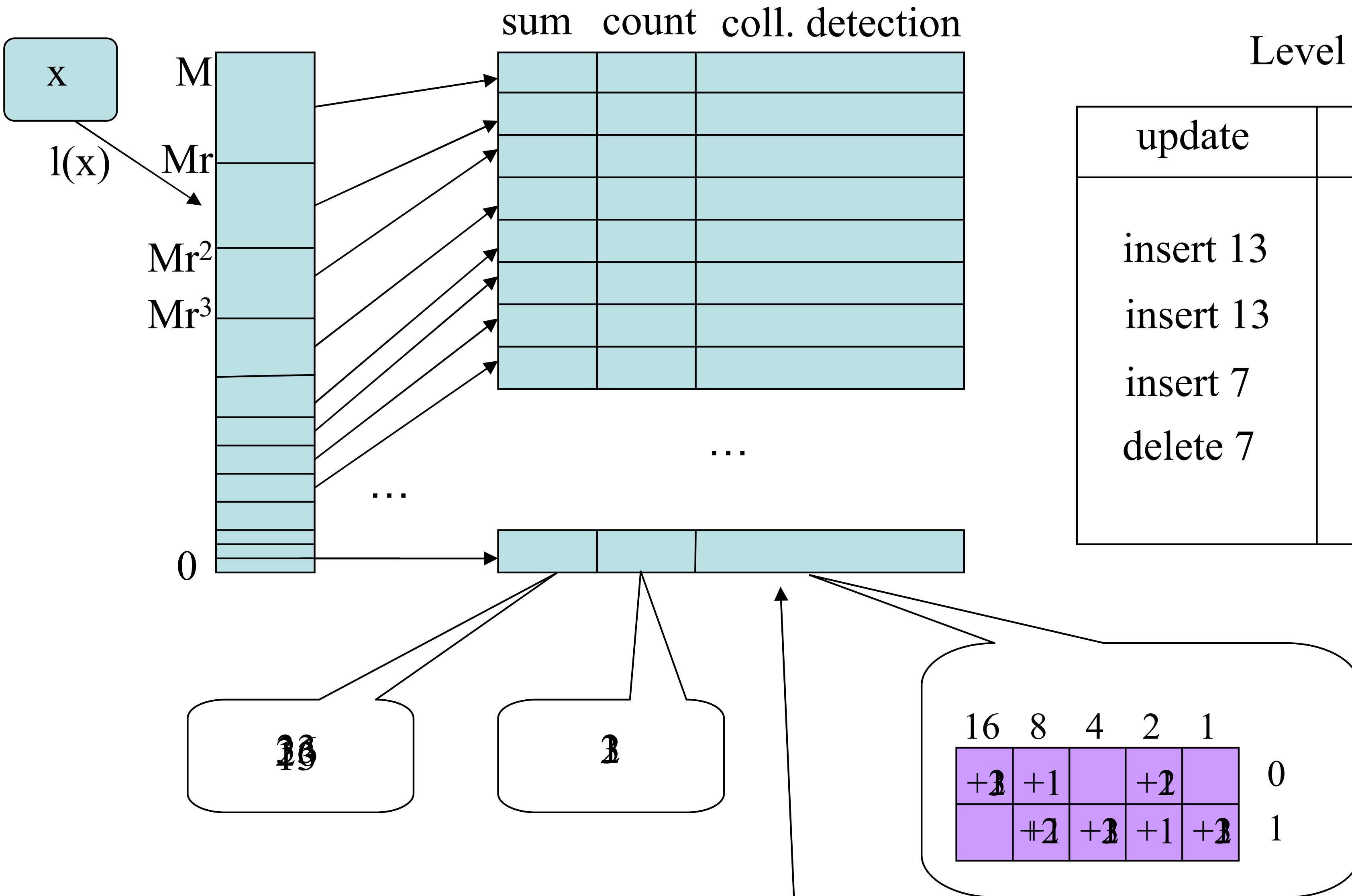
Track the following information as updates are seen:

- x: Item with largest hash value seen so far
- unique: Is it the only distinct item seen with that hash value?
- count: Count of the item x

Easy to keep (x, unique, count) up to date for insertions only

Challenge:
How to maintain in presence of deletes?

Collision Detection: inserts and deletes



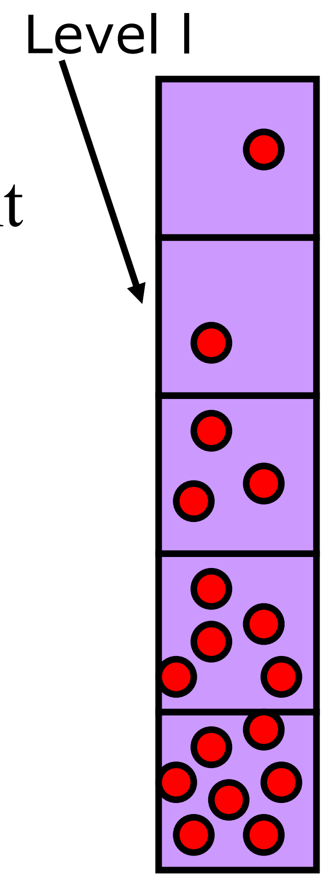
Level 0

update	output
insert 13	13/1=13
insert 13	26/2=13
insert 7	collision
delete 7	26/2=13

Simple: Use approximate distinct element estimation routine.

Outline of Analysis

- Analysis shows: if there's unique item, it's chosen uniformly from set of items with non-zero count.
- Can show whatever the distribution of items, the probability of a unique item at level l is at least constant
- Use properties of hash function:
  - only limited, pairwise independence needed (easy to obtain)
- Theorem: With constant probability, for an arbitrary sequence of insertions and deletes, the procedure returns a uniform sample from the inverse distribution with constant probability.
- Repeat the process independently with different hash functions to return larger sample, with high probability.



Application to Inverse Distribution Estimates

Overall Procedure:

- Obtain the distinct sample from the inverse distribution of size s;
- Evaluate the query on the sample and return the result.
  - Median number of bytes sent: find median from sample
  - The most common volume of traffic sent: find the most common from sample
  - What fraction of items sent i bytes: find fraction from the sample

Example:

- Median is bigger than ½ and smaller than ½ the values.
- Answer has some error: not ½, but $(½ \pm \varepsilon)$

Theorem: If sample size $s = O(1/\varepsilon^2 \log 1/\delta)$ then answer from the sample is between $(½-\varepsilon)$ and $(½+\varepsilon)$ with probability at least $1-\delta$.

Proof follows from application of Hoeffding's bound.

Experimental Study

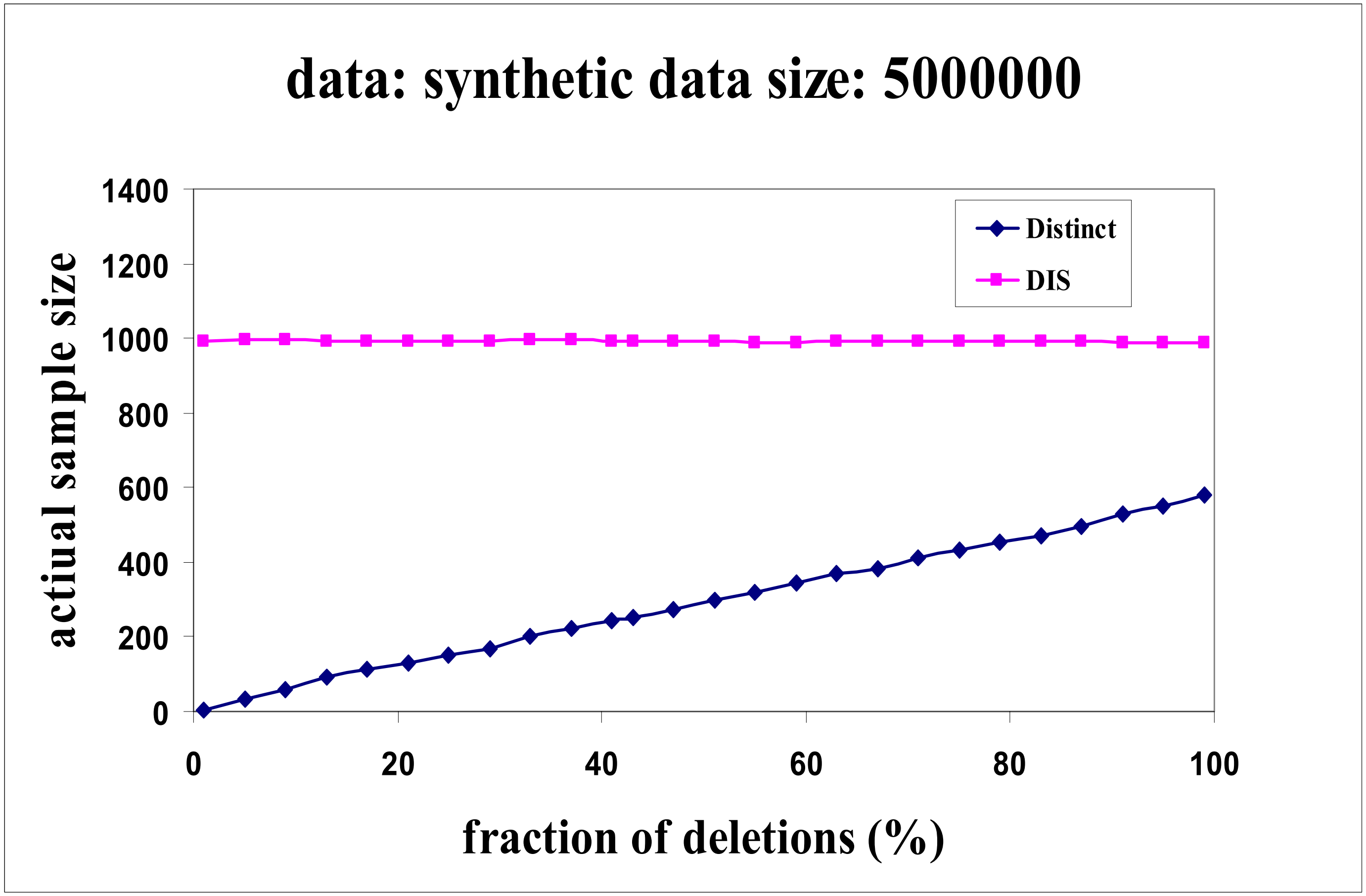
Data sets:

- Large sets of network data drawn from HTTP log files from the 1998 World Cup Web Site (several million records each)
- Synthetic data set with 5 million randomly generated distinct items
- Used to build a dynamic transactions set with many insertions and deletions

- **(DIS) Dynamic Inverse Sampling algorithms** – extract at most one sample from each data structure
- **(GDIS) Greedy version of Dynamic Inverse Sampling** – greedily process every level, extract as many samples as possible from each data structure
- **(Distinct) Distinct Sampling (Gibbons VLDB 2001)** draws a sample based on a coin-tossing procedure using a pairwise-independent hash function on item values

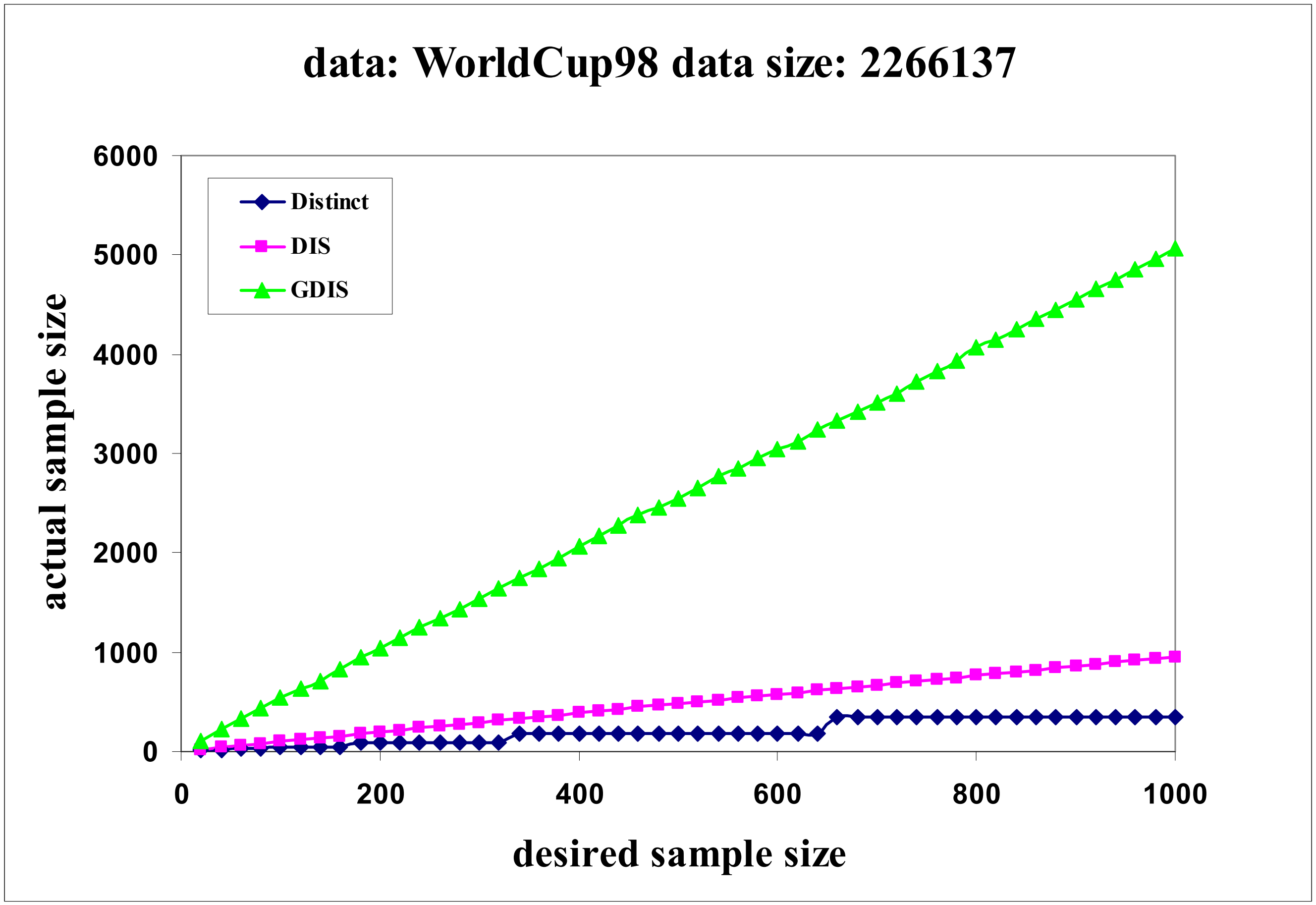
Sample Size vs. Fraction of Deletions

Desired sample size is 1000.



Returned Sample Size

Experiments were run on the client ID attribute of the HTTP log data. 50% of the inserted records were deleted.



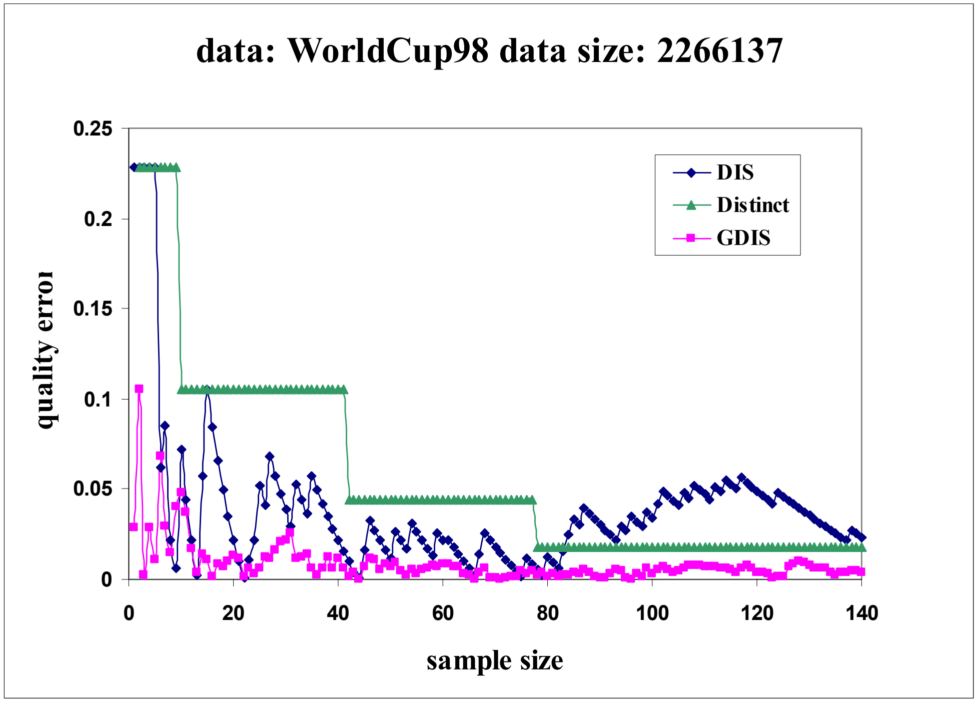
Sample Quality

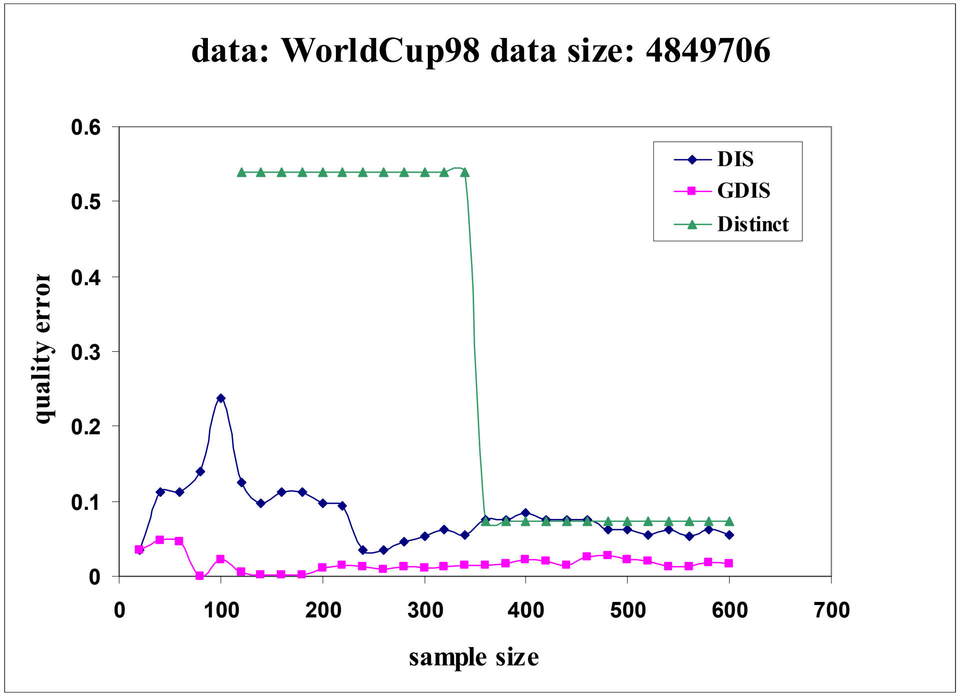
Inverse range query:

Compute the fraction of records with size greater than i=1024 and compare it to the exact value computed offline

Inverse quantile query:

Estimate the median of the inverse distribution using the sample and measure how far was the position of the returned item i from 0.5.





Related Work

- Distinct sampling under insert only:
  - Gibbons: Distinct Sampling, VLDB 2002.
  - Datar and Muthukrishnan: Rarity and similarity, ESA 2002.
- Distinct sampling under deletes also:
  - Frahling, Indyk, Sohler: Dynamic geometric streams, STOC 2005.
  - Ganguly, Garofalakis, Rastogi: Processing Set Expressions over Continuous Update Streams, SIGMOD 2003.
- Inverse distributions:
  - Has recently informally appeared in networking papers.

Conclusions

- We have formalized Inverse Distributions on data streams and introduced Dynamic Inverse Sampling method that draws uniform samples from the inverse distribution in presence of insertions and deletions.

- With a sample of size $O(1/\varepsilon^2)$, can answer many queries on the inverse distribution (including point and range queries, heavy hitters, quantiles) up to additive approximation of ε.

- Experimental study shows that proposed methods can work at high rates and answer queries with high accuracy

- Future work:
  - Incorporate in data stream systems
  - Can we also sample from forward dbn under inserts and deletes?