Tracking Inverse Distributions of Network Data Streams and Applications

Graham Cormode
cormode@bell-labs.com

S. Muthukrishnan
muthu@cs.rutgers.edu

Lucent Technologies
Bell Labs Innovations

Motivating Problems

INV – How many people made less than five VoIP calls today?

FWD – Which are the most frequently called numbers?

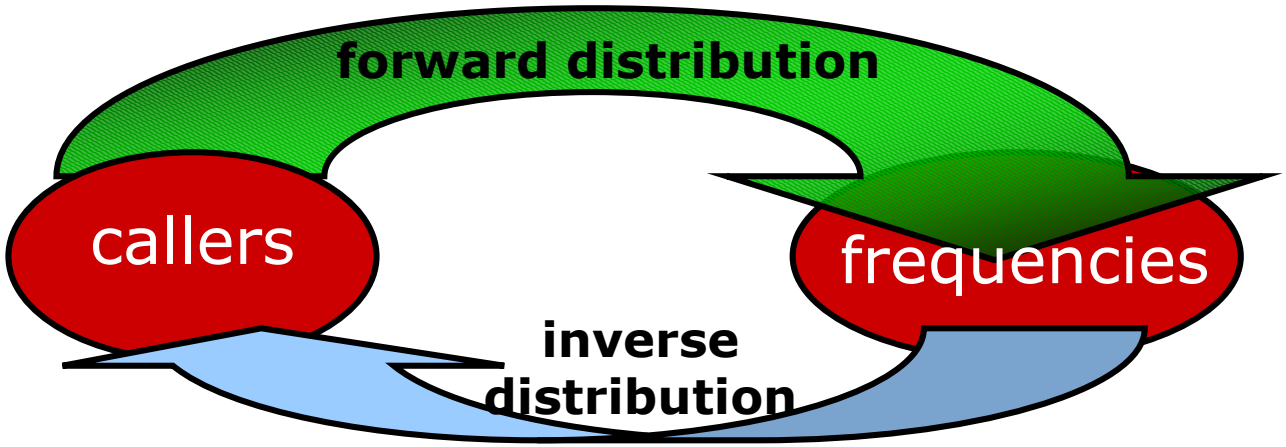
INV – What is most frequent number of calls made?

FWD – What is median call length?

INV – What is median number of calls?

FWD – How many calls did subscriber S make?

Can classify these questions into two types: questions on the *forward distribution* and on the *inverse distribution*.



The Inverse Distribution

- Forward distribution $f[0\ldots U]$,
  $f(x)$ = number of calls / bytes / packets etc.

  How many calls did S make? Find $f(S)$
  Most frequently caller? Find x s.t. $f(x)$ is greatest

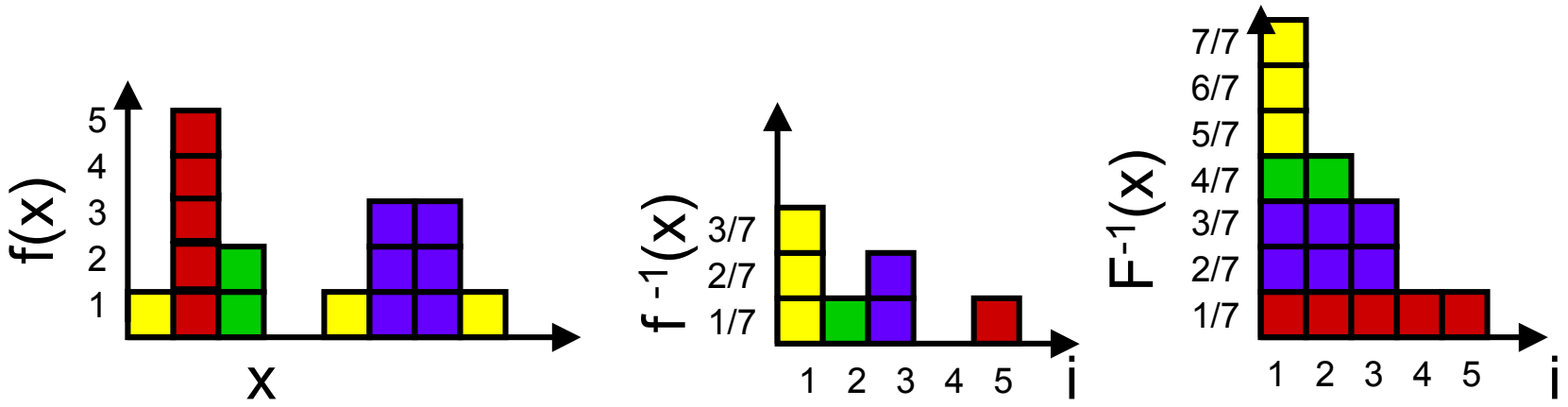
- Inverse distribution is $f^{-1}[0\ldots N]$,
  $f^{-1}(i)$ = fraction of users making i calls.
  $= |\{ x : f(x) = i, i \neq 0\}/|\{x : f(x) \neq 0\}|$

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

  Number of people making < 5 calls $= 1 - F^{-1}(5)$
  Most common number of calls made = i s.t. $f^{-1}(i)$ is greatest

- In linear space, easy to go from forward to inverse dbn.
  Much more difficult in small space given data presented in forward dbn to extract inverse dbn, little prior work.

Examples



Separation between tracking inverse dbn and forward dbn: consider tracking a simple point query on each distribution. Eg. Find f(9085827700): count calls involving this party

But finding $f^{-1}(2)$ is provably hard: can't track exactly how many people made 2 calls without keeping full space

Even approximating up to some constant factor is hard.

We show how to sample from inv dbn and use the sample.

Sampling Insight

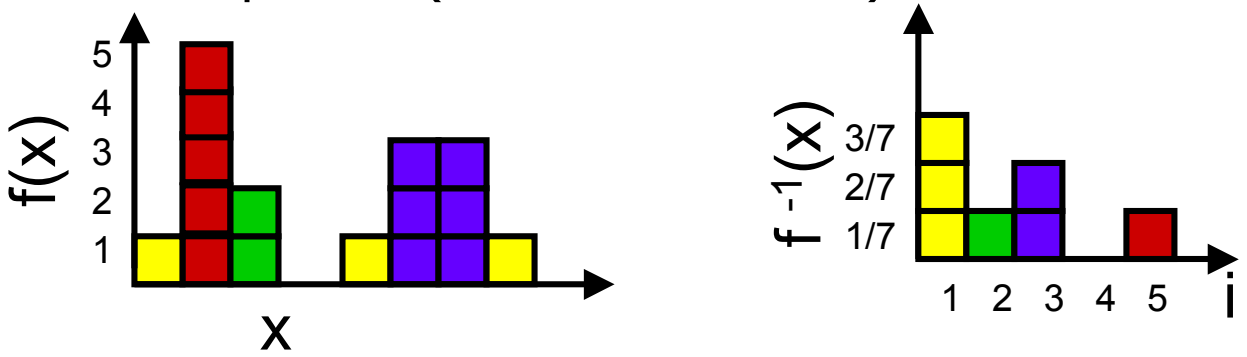
See a stream of items x. Count of x is f(x) = i.

Each distinct item x contributes to one pair (i,x)
Need to sample uniformly from these pairs.

Basic insight: sample uniformly from the items x and count how many times x is seen to give (i,x) pair that has correct i and is uniform.

How to pick x uniformly from those with non-zero count?

Use a randomly chosen hash function on each x to decide whether to pick it (and reset count).



Hashing Technique

Use hash function h with exponentially decreasing distribution:

$\Pr[h(x) = l] = r^l(1-r)$ r is an appropriate const < 1

Track the following information as updates are seen:

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

Easy to keep (x, uniq, count) up to date as new items arrive

Theorem: If uniq is true, then x is picked uniformly. Probability of uniq being true is at least a constant.

Proof outline: Uniformity follows so long as hash function h is at least pairwise independent. Hard part is showing that uniq is true with constant prob.

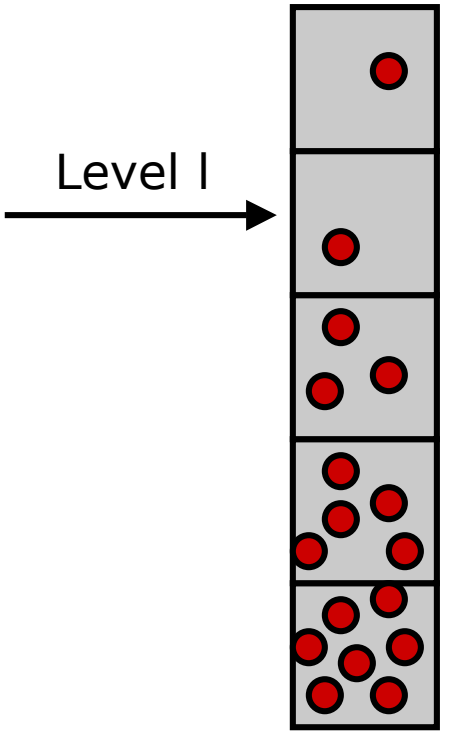
Hashing analysis

If only one item at level l, then uniq is true

If two items at level l or higher, can go deeper into the analysis and show that (assuming there are two items) there is constant probability that they are both at same level.

If not at same level, then uniq is true, and we recover a uniform sample.

- Probability of failure is p = r(3+r)/(2(1+r)).
- Number of levels is O(log N / log 1/r)
- Need 1/r > 1 so this is bounded, and $1/r^2$ ¸ 3/2 for analysis to work
- End up choosing r = p(2/3), so p is < 1



Using the Sample

Repeat sufficiently many times to draw a sample from the inverse distribution. Sample of size s can be used for a variety of problems with guaranteed accuracy.

! Evaluate the question of the sample and return the result.

Eg. Median number of calls made: find median from sample

Median is bigger than ½ and smaller than ½ the values.

Answer has some error: not ½, but $(½ \S \varepsilon)$

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.

Sampling From the Difference

How to compare two streams and look at their difference.
Eg.: what's the difference between yesterday and today; what's the difference between Router A and Router B etc.

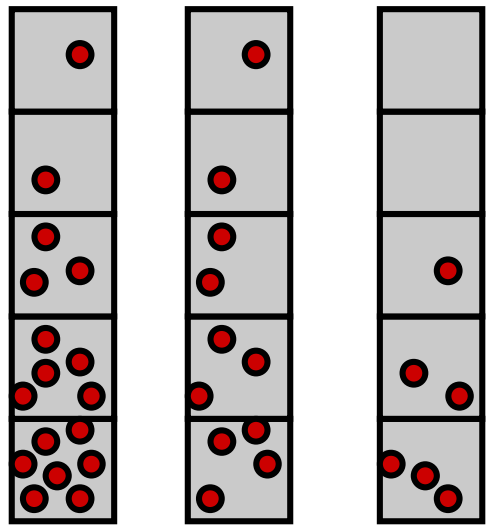
The *difference distributions:* $(f-g)(x) = f(x) - g(x)$ and $(f-g)^{-1}$

Can take the hashing approach, and combine two summaries to get summary of difference in inv dbn.

Sample (i,x) uniformly from $(f-g)$ so x is chosen uniformly from x where $(f-g)(x) \neq 0$.

Idea: track info about all levels. Ensure when combining two synopses result is uniform over $(f-g)^{-1}$

Ensure that combining info about f and g has duplicate items exactly canceling out.



A Potential Application…

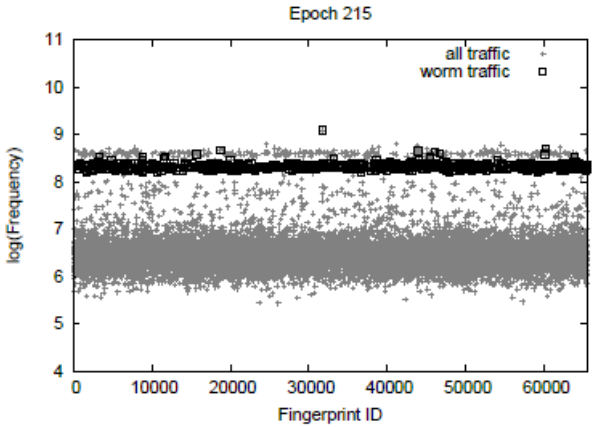
Inverse distribution can be applied to detecting new attacks

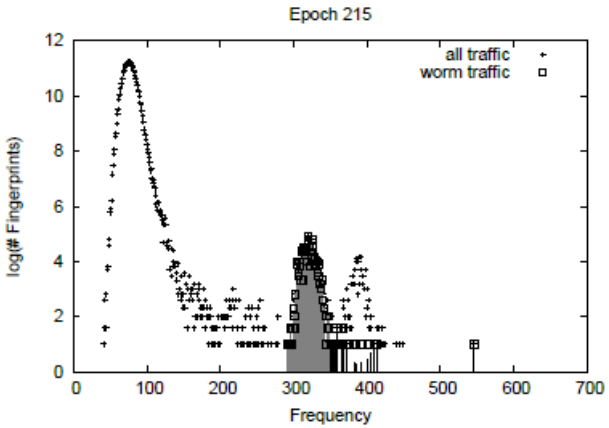
Look at forward distribution of substrings in packet content: New worms manifest as high values in forward distribution.

But many peaks in normal traffic, need to filter false alarms

Looking at the inverse distribution, we see worms much earlier as "bumps" in the distribution.

These "bumps" move "up" inverse dbn as worm spreads, ie are significant in difference in inverse distribution.





Karamcheti, Geiger, Kedem, Muthukrishnan 2005