Affirmative Sampling

Conrado Martínez
Univ. Politècnica de Catalunya, Barcelona, Spain

AofA'19, CIRM Luminy
June 2019

Joint work with:

Jérémie Lumbroso (Princeton)

Introduction

- A data stream is a (very long) sequence

$$\mathcal{Z} = z_1, z_2, z_3, \ldots, z_N$$

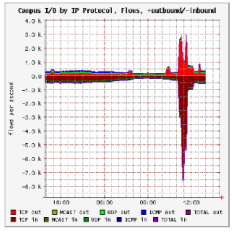
  of items z_i drawn from some (large) domain $\mathcal{U}$, $z_i \in \mathcal{U}$
- The goal: to extract information from $\mathcal{Z}$, but ...

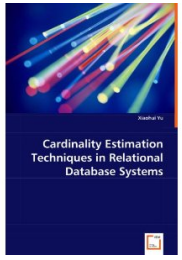
Introduction

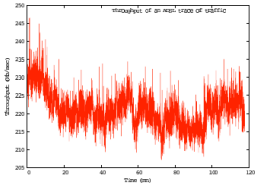
. . . there are limitations to our computational power:

- a single pass over the sequence
- very short time for computation on each item
- very small auxiliary memory: $M \ll N$; ideally $M = \Theta(1)$ or $M = \mathcal{O}(\log N)$
- no statistical hypothesis on the data

Introduction







There are lots of applications for this data stream model:

- Network traffic analysis $\Rightarrow$ DoS/DDoS attacks, worms, ...
- Database query optimization
- Information retrieval $\Rightarrow$ similarity index
- Data mining
- And many more ...

Introduction

We will often see $\mathcal{Z}$ as a multiset

$$\{x_1^{f_1}, \ldots, x_n^{f_n}\},$$

with

$$x_i = \text{the } i\text{-th distinct element}$$
$$f_i = \text{frequency of } x_i$$

Introduction

Some typical problems:

- Find the cardinality of $\mathcal{Z}$: card$(\mathcal{Z}) = n \leqslant N$
- Find the top-k most frequent elements
- Find c-icebergs (a.k.a. *heavy hitters*): $f_i/N > c$
- Draw a random sample of k elements z_i from the stream (with Prob $= 1/N$)
- Draw a random sample of k distinct elements x_i (with Prob $= 1/n$)

Distinct Sampling

procedure DISTINCTSAMPLING(k, $\mathcal{Z}$)
    fill S with the first k distinct elements (and hash values)
    of the stream $\mathcal{Z}$
    for all $z \in \mathcal{Z}$ **do**
        if HASH(z) $<$ min hash value in S **then**
            Discard z; **continue**
        ▷ HASH(z) $>$ min hash value in S
        if $z \in S$ **then**
            Update z stats
        else ▷ replace elem of min. hash with z
            $S \leftarrow S \setminus \{\text{elem. with min. hash in } S\} \cup \{z\}$
    return S

Distinct Sampling

- The algorithm draws a random sample of k distinct elements (each one has prob. $1/n$ of being drawn), by keeping in the sample the k elements with the largest hash values seen so far[1]
- If we use uniform random numbers in $(0, 1)$ instead of hash values (and don't check if $z \in S$) $\Rightarrow$ Reservoir Sampling

[1] Pragmatic assumptions: hash values uniformly distributed; probability of collisions negligible

Distinct Sampling

- Adaptive Sampling (Wegman, 1984 & Flajolet, 1990) delivers a sample of at most $\leqslant \mathrm{B}$ distinct elements for a fixed cache size B + cardinality estimation using the size of the sample
- But . . . what if we want a sample with size depending on (and growing with!) $\mathfrak{n}$?

Affirmative Sampling

- The larger the cardinality (n) the larger the samples $\Rightarrow$ samples better represent diversity
- All distinct elements have the same opportunity to be sampled

Affirmative Sampling

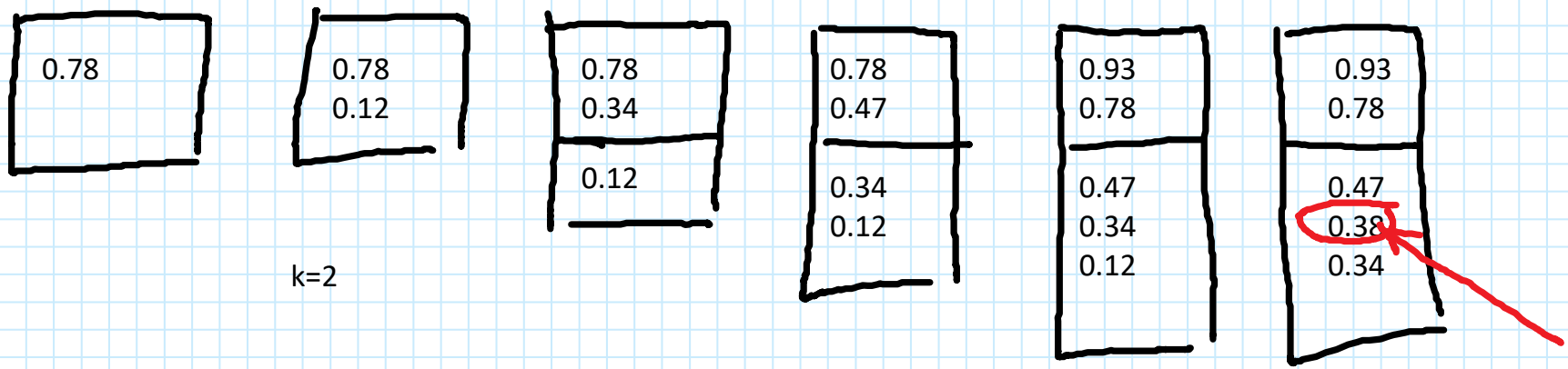
- The larger the cardinality (n) the larger the samples $\Rightarrow$ samples better represent diversity
- All distinct elements have the same opportunity to be sampled

Affirmative Sampling!!

Affirmative Sampling

procedure AFFIRMATIVESAMPLING(k, $\mathcal{Z}$)
    fill S with the first k distinct elements (and hash values)
    of the stream $\mathcal{Z}$
    for all $z \in \mathcal{S}$ **do**
        if HASH(z) $<$ min hash value in S **then**
            Discard z; **continue**
        ▷ HASH(z) $>$ min hash value in S
        if $z \in S$ **then**
            Update z stats
        else if HASH(z) $>$ k-th largest hash value in S **then**
            $S \leftarrow S \cup \{z\}$
        else ▷ replace elem of min. hash with z
            $S \leftarrow S \setminus \{$elem. with min. hash in $S\} \cup \{z\}$
    return S

Affirmative Sampling



0.78 0.12 0.34 0.12 0.47 0.06 0.93 0.38...

Affirmative Sampling

- The size of the sample S is a random variable = the number of k-records in a random permutation of size n
- The sample does not contain the k-records, but the $|S|$ elements with the largest hash values seen so far $\Rightarrow$ S is a random sample
- If $x \in S$ then x has been added to S in its very first occurrence and it has remained in S ever since $\Rightarrow$ can collect exact stats (e.g. frequency counts) for x

Affirmative Sampling

- Properties of $|S|$ are very well understood; in particular

$$\mathbb{E}\{|S|\} = k \ln(n/k) + \text{l.o.t.}$$

  The exact and asymptotic distribution of R, moments, ... is known (e.g., Helmi, M., Panholzer, 2014)
- Estimating cardinality (RECORDINALITY, Helmi, Lumbroso, M., Viola, 2012)

$$\mathbb{E}\left\{k\left(1+\frac{1}{k}\right)^{|S|-k+1}-1\right\} = n$$

Affirmative Sampling

- We also understand fairly well F = number of times an element substitutes another in the sample (not a k-record, but larger than some k-record):

$$\mathbb{E}\{F_n\} = k \ln^2(n/k) + \text{l.o.t.}$$

- Expected cost of Affirmative Sampling

$$\begin{aligned}\mathbb{E}\{C\} &= \Theta(N + (\mathbb{E}\{|S|\} + \mathbb{E}\{F\}) \log \mathbb{E}\{|S|\}) \\ &= \Theta(N + (\log^2 n) \cdot (\log \log n))\end{aligned}$$

using hashing for membership and a couple of priority queues (one of fixed size k, the other of size $|S| - k$)

Affirmative Sampling & Inference

Examples of the queries we might be interested in:

- What fraction of the distinct elements have relative frequency below 5%? ($\mathrm{f_i}/\mathrm{N} < 0.05$)
- How many distinct element have we seen during the last 10 minutes?

Affirmative Sampling & Inference

- Larger samples mean more accurate inferences. If n_P is the number of distinct elements satisfying some property P then

$$\mathbb{E}\left\{\frac{\#\text{ elems in } S \text{ satisfying } P}{|S|}\right\} = \frac{n_P}{n}$$

- This is true whatever the size of S (even if fixed), but

$$\mathbb{V}\left\{\frac{\#\text{ elems in } S \text{ satisfying } P}{|S|}\right\} \sim \frac{n_p}{n} \cdot \left(1 - \frac{n_p}{n}\right) \cdot \mathbb{E}\left\{\frac{1}{|S|}\right\},$$

  and $\mathbb{E}\{1/|S|\} \sim 1/\mathbb{E}\{|S|\} = 1/(k \ln(n/k))$

- $\Rightarrow$ Standard error decreases as

$$1/\sqrt{\log n} = 1/\sqrt{\text{“size of the memory”}}$$

Affirmative Sampling & Applications

- Inference not only about proportions, but also accurate estimates of n_P
- Accurate estimators of similarity index (Jaccard's index) between two sets/sequences
- Good approximations of quantiles, e.g., get an element x with relative rank $0.5n \pm o(n)$
- Very accurate (and independent cardinality estimation) via $|S|$-th largest hash value in the sample
- Better detection of rare events (outliers) than drawing random samples from the stream

Variants

- Instead of adding new elements to the sample when the hash value is among the largest k seen so far, one can add an element whenever its hash value is above the top $(100 \cdot \alpha)$% largest hash values (e.g., above the median) so far. $\Rightarrow \mathbb{E}\{|S|\} = \Theta(n^{\alpha})$ instead of $\sim k \ln(n/k)$
- Also: $\mathbb{E}\{1/|S|\} = \Theta(n^{-\alpha}) \Rightarrow$ increased accuracy, variance goes faster to 0 for large n
- The size of S has also been well studied and analyzed for that rule (Krieger et al, 2008; Archibald, M. 2009; Helmi, Panholzer, 2013; Gaither, 2012; Janson, 2019)

Open problems

- Samples of expected size $= f(n)$ for given target sublinear function f?
- How to make $\mathbb{V}\{|S|\}$ as small as possible?
- Affirmative sampling in the sliding window model? (we know how to do distinct sampling for fixed size samples)


Thanks a lot
for your attention!