Computational Frameworks

Streaming (Part 2)

OUTLINE

1. Sketching
   - Estimating individual frequencies
   - Estimating the second moment F_2
2. Filtering: Bloom filters for membership problem

Sketching

Objective

Let us recall **our setting:**

- Stream $\Sigma = x_1, x_2, \ldots, x_n$, whose elements belong to a universe U, with $|U| = M$.
- For each $u \in U$ its frequency in Σ is

$$f_u = |\{j : x_j = u, 1 \leq j \leq n\}|,$$

In one pass over Σ we want to compute a **small sketch** which enables to compute **unbiased estimates** of

- f_u for any given $u \in U$ (*individual frequencies*)
- $F_2 = \sum_{u \in U}(f_u)^2$ (*second moment*)

which exhibit provable (probabilistic) space-accuracy tradeoffs

Observation: clearly, the exact computation of all f_u's or of F_2 might require space proportional to $|\Sigma|$.

Count-min sketch

The first approach we consider is based on the count-min sketch invented by [Cormode,Muthukrishnan 2003].

Main ingredients

- $d \times w$ array C of counters ($O(\log n)$ bits each)
- d hash functions: $h_1, h_2, \ldots, h_d$, with

$$h_j \ : \ U \to \{1, 2, \ldots, w\},$$

  for every j.

Note that d and w are *design parameters* that regulate the space/time-accuracy tradeoff.

Count-min sketch: algorithm

Initialization: $C[j,k] = 0$, for every $1 \leq j \leq d$ and $1 \leq k \leq w$.

For each x_t in Σ **do**

For $1 \leq j \leq d$ **do** $C[j, h_j(x_t)] \leftarrow C[j, h_j(x_t)] + 1$;

At the end of the stream: for any $u \in U$, its frequency f_u can be estimated as:

$$\tilde{f}_u = \min_{1 \leq j \leq d} C[j, h_j(u)].$$

It is always true that $\tilde{f}_u \geq f_u \quad \forall u$

$\Rightarrow$ biased estimator

Example: $n = 15, d = 3, w = 3$

$\Sigma = A, B, C, B, D, A, C, D, A, B, D, C, A, A, B$

u, f_u	h_1	h_2	h_3
A, 5	1	2	2
B, 4	2	3	2
C, 3	1	1	3
D, 3	2	2	3

Array C

	1	2	3
1	$5_A + 3_C$	$4_B + 3_D$	
2	3_C	$5_A + 3_D$	4_B
3		$5_A + 4_B$	$3_C + 3_D$

- $\tilde{f}_A = \min\{8, 8, 9\} = 8 > f_A = 5$
- $\tilde{f}_B = \min\{7, 4, 9\} = 4 = f_B$
- $\tilde{f}_C = \min\{8, 3, 6\} = 3 = f_C$
- $\tilde{f}_D = \min\{7, 8, 6\} = 6 > f_D$

Count-min sketch: analysis

We assume:

- the d hash functions $h_1, h_2, \ldots, h_d$ are mutually independent
- for each $j \in [1, d]$ and each $u, v \in U$ with $u \neq v$, $h_j(u)$ and $h_j(v)$ are independent random variables uniformly distributed in $[1, w]$.

Theorem

Consider a $d \times w$ count-min sketch for a stream Σ of length n, where $d = \log_2(1/\delta)$ and $w = 2/\epsilon$, for some $\delta, \epsilon \in (0, 1)$. The sketch ensures that for any given $u \in U$ occurring in Σ

accuracy

$$\tilde{f}_u - f_u \leq \epsilon \cdot n,$$

with probability $\geq 1 - \delta$. confidence

Obs.: The bias in the estimated frequencies discourages their use to estimate the second moment F_2.

Intuition

if $h_j(v)$ is independent of $h_j(u)$ for $u \neq v$ then $C[j, h_j(u)]$ will receive, on average a fraction w of the distinct elements of Σ, hence a fraction w of the sum of all frequencies i.e. n/w

$\Rightarrow$ Together with f_u cell $C[j, h_j(u)]$ will contain an additional contribution which is, on average, $\leq n/w$

$\Rightarrow$ if $w = 2/\varepsilon$ then $n/w = \varepsilon n/2$

$\Rightarrow$ By Markov inequality

$C[j, h_j(u)] - f_u \leq \varepsilon n$

with constant probability

Then repeating d times, the probability becomes $1-\delta$ using $d = O(\log(1/\delta))$

Count sketch

The count sketch was invented by [Charikar,Chen,Farach-Colton 2002], and can be seen as an **unbiased variant of the count-min sketch**.

IDEA: for each item $u \in U$ multiply its contributions to each row by a value in $\{-1, +1\}$ randomly selected, so to cancel out collisions.

Main ingredients

- $d \times w$ array C of counters ($O(\log n)$ bits each)
- d hash functions: $h_1, h_2, \ldots, h_d$, with

$$h_j : U \to \{1, 2, \ldots, w\},$$

  for every j.
- d hash functions: $g_1, g_2, \ldots, g_d$, with

$$g_j : U \to \{-1, +1\},$$

  for every j.

Count sketch: algorithm

Initialization: $C[j,k] = 0$, for every $1 \leq j \leq d$ and $1 \leq k \leq w$.

For each x_t in Σ **do**

For $1 \leq j \leq d$ **do** $C[j, h_j(x_t)] \leftarrow C[j, h_j(x_t)] + g_j(x_t)$;

At the end of the stream: for any $u \in U$ and $1 \leq j \leq d$, let

$$\tilde{f}_{u,j} = g_j(u) \cdot C[j, h_j(u)].$$

The frequency of u can be estimated as:

$$\tilde{f}_u = \text{median of the } \tilde{f}_{u,j}\text{'s}$$

Example: $n = 15, d = 3, w = 3$

$\Sigma = A, B, C, B, D, A, C, D, A, B, D, C, A, A, B$

u, f_u	h_1	g_1	h_2	g_2	h_3	g_3
A, 5	1	+1	2	+1	2	+1
B, 4	2	-1	3	+1	2	-1
C, 3	1	-1	1	-1	3	+1
D, 3	2	-1	2	+1	3	+1

Array C		
$5_A - 3_C$	$-4_B - 3_D$	
-3_C	$5_A + 3_D$	4_B
	$5_A - 4_B$	$3_C + 3_D$

- $\tilde{f}_A = \text{median}\{2, 8, 1\} = 2 < f_A = 5$
- $\tilde{f}_B = \text{median}\{7, 4, -1\} = 4 = f_B$
- $\tilde{f}_C = \text{median}\{-2, 3, 6\} = 3 = f_C$
- $\tilde{f}_D = \text{median}\{7, 8, 6\} = 6 > f_D = 3$

Count sketch: analysis

Assumptions: for both sets of hash functions (the h_j's and the g_j's) we make the same assumptions of independence and uniform distribution, which we made for the h_j's in the analysis of the count-min sketch.

Theorem

Consider a $d \times w$ count-~~min~~ sketch for a stream Σ of length n, where $d = \log_2(1/\delta)$ and $w = O\left(1/\epsilon^2\right)$, for some $\delta, \epsilon \in (0,1)$. The sketch ensures that for any given $u \in U$ occurring in Σ:

- $\mathrm{E}[\tilde{f}_{u,j}] = f_u$, *for any $j \in [1,d]$, i.e., $\tilde{f}_{u,j}$ is an unbiased estimator of f_u;*
- *With probability $\geq 1 - \delta$,*

$$|\tilde{f}_u - f_u| \leq \epsilon \cdot \sqrt{F_2},$$

where $F_2 = \sum_{u \in U}(f_u)^2$ (true second moment).

Intuition. Due to the random signs, on average the "noise" created by several items colliding on the same colum as a u, cancel out.

Count-min sketch vs count sketch

Guaranteed discrepancy for

count-min sketch $\quad \varepsilon n$

count sketch $\quad \varepsilon \sqrt{F_2}$

$$F_2 = \sum_{u \in U} (f_u)^2 \leq \left(\sum_{u \in U} f_u \right)^2 = n^2$$

$$\varepsilon \sqrt{F_2} \leq \varepsilon \sqrt{n^2} = \varepsilon n$$

$\Rightarrow$ count-sketch provides unbiased estimates (in each row) and tighter $\tilde{f}_u$'s at least for not too skewed distribution, but needs more space

Estimation of F_2

Given a $d \times w$ count-~~min~~ sketch for Σ, define

$$\tilde{F}_{2,j} = \sum_{k=1}^{w} (C[j,k])^2. \qquad \forall\ 1 \le j \le d$$

We can derive the following estimate for the true second moment F_2:

$$\tilde{F}_2 = \text{median of the } \tilde{F}_{2,j}\text{'s}$$

Observation: the estimator is the result of a line of resarch initiated by [Alon, Matias, Szegedy 1996] who proposed the AMS sketch, and ancestor of the count sketch.

Example (same as before)

$\Sigma = A, B, C, B, D, A, C, D, A, B, D, C, A, A, B$

$F_2 = (f_A)^2 + (f_B)^2 + (f_C)^2 + (f_D)^2 = 5^2 + 4^2 + 3^2 + 3^2 = 59.$

Array C from before		
2	-7	
-3	8	4
	1	6

- Estimate from row $j = 1$: $2^2 + (-7)^2 = 53$
- Estimate from row $j = 2$: $(-3)^2 + 8^2 + 4^2 = 89$
- Estimate from row $j = 3$: $1^2 + 6^2 = 37$

$\Rightarrow \tilde{F}_2 = 53 \qquad (F_2 = 59)$

Analysis of $\tilde{F}_2$

The following theorem can be proved under the same assumptions made for the analysis of the count sketch

Theorem

Consider a $d \times w$ count-min sketch for a stream Σ of length n, where $d = \log_2(1/\delta)$ and $w = O\left(1/\epsilon^2\right)$, for some $\delta, \epsilon \in (0,1)$. The sketch ensures that:

- $\mathrm{E}[\tilde{F}_{2,j}] = F_2$, *for any $j \in [1, d]$, i.e., $\tilde{F}_{2,j}$ is an unbiased estimator of F_2;*
- *With probability $\geq 1 - \delta$,*

$$|\tilde{F}_2 - F_2| \leq \epsilon \cdot \sqrt{F_2},$$

In the following slides, we show that $\mathrm{E}[\tilde{F}_{2,j}] = F_2$ for every j, while we skip the proof of the second bullet point.

Analysis of performance metrics

Both count-min and count sketches can be computed in **1 pass**

To assess space and time performance, we assume:

- Each hash function can be applied in constant time
- The space occupied by the sketch dominates over the one needed to store the hash functions

For both sketches we have

- **Working memory:** $O(d \cdot w)$, which becomes $O(\log(1/\delta)/\epsilon)$, for the count-min sketch, and $O\left(\log(1/\delta)/\epsilon^2\right)$, for the count sketch, in order to attain the proabilistic accuracy stated before.
- **Processing time per element:** $O(d) = O(\log(1/\delta))$,

Moreover, given the sketch, the estimates $\tilde{f}_u$'s (individual frequencies) and $\tilde{F}_2$ (second moment) can be computed in $O(d)$ and $O(d \cdot w)$ time, respectively.

Filtering

Motivation

For many applications, processing a data stream $\Sigma = x_1, x_2, \ldots$ entails essentially the identification of the x_i's which meet a certain criterion.

Some criteria can be checked very easily with a minimum cost in terms of space and time. However, this is not always the case.

Example. Suppose that the x_i's are email addresses and that when x_i arrives we need to check whether it belongs to a set S of *verified* addresses. If S is very large (e.g., 1 billion addresses of approximately 20 bytes each), we face two issues:

- If S does not fit into main memory, it must be stored on disk.
- Standard exact techniques to check $x_i \in S$, especially if S is on disk, may be time consuming and not compatible with a high arrival rate.

Can we check membership efficiently with reasonable accuracy?

Bloom filter

Approximate membership problem

Given a stream $\Sigma = x_1, x_2, \ldots$ of elements from some universe U, and let S be a set of m elements from U. Store S into a compact data structure that, for any given x_i, allows to check whether $x_i \in S$ with

- no error, when $x_i \in S$ (No false negatives)
- small probability error, when $x_i \notin S$ (Small false positive rate)

A solution to the problem comes from the **Bloom filter**, introduced in [Bloom 1970]. Its **main ingredients** are:

- Array A of n bits, all initially 0.
- k hash functions: $h_1, h_2, \ldots, h_k$, with

$$h_j \ : \ U \to \{0, 1, \ldots, n-1\} \qquad \text{for every } 1 \leq j \leq k$$

Note that n and k are *design parameters* that regulate the tradeoff between space/time and accuracy.

Bloom filter

Initialization:

For each $e \in S$ **do**

For $1 \leq j \leq k$ **do** $A[h_j(e)] \leftarrow 1$;

A is the compact representation of S we are looking for

Membership test: for any x_i in Σ if

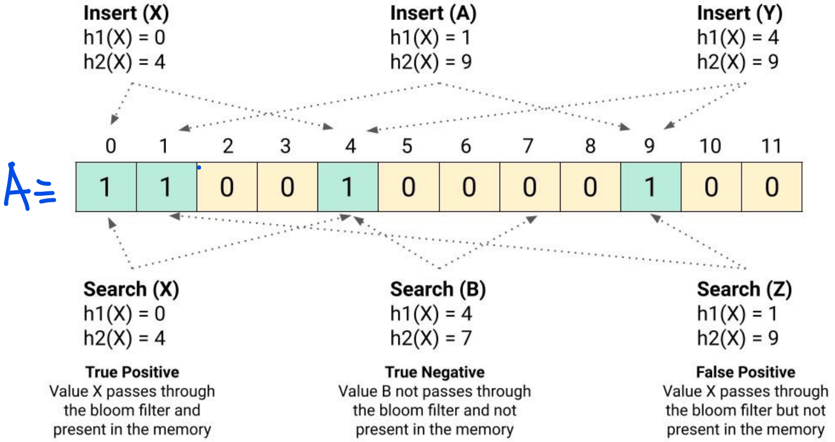
$$x_i \in S \quad \Leftrightarrow \quad h_1(x_i) = h_2(x_i) = \cdots = h_k(x_i) = 1$$

Straighforward properties:

- The approach ensures that there are no false negatives
- Assuming that $k \ll n$, and that the hash functions can be stored compactly, the required working memory is dominated by the storage of $A \Rightarrow n$ bits.
- Assuming that each hash function can be applied in $O(1)$ time, the membership test requires $O(k)$ time.

Example

$S = \{x, A, y\}$ $k = 2$



Bloom filter: analysis of false positive rate

Assumptions: for the set of hash functions (the h_j's) we make the same assumptions of independence and uniform distribution, which we made in the analysis of the count-min sketch.

Theorem

Suppose that n is ufficiently large. For any given x_i which does not belong to S, the probability that x_i is erroneously claimed to be in S is

$$\Pr(A[h_j(x_i)] = 1 \text{ for each } 1 \leq j \leq k) \simeq (1 - e^{-km/n})^k$$

This probability is referred to as false positive rate.

Email example: In the case of email addresses mentioned before, $m = 10^9$ and storing the entire set S would require 20GB (assuming that each email takes 20 bytes). Using a Bloom filter with $n = 8m$ (hence $|A| = 1\text{GB}$), and $k = 6$, the false positive rate is about 2.15%.

Let $S = \{ e_1\ e_2 \dots e_m \}$

With the usual assumptions on independence and uniform distribution for the hash functions the indices of the 1's can be seen as $k \cdot m$ independent and uniformly distributed variables in $[0 \div n-1]$

$$\text{Prob}(A[l] = 0) = \left(1 - \frac{1}{n}\right)^{k \cdot m}$$

for arbitrary l

$$= \left(1 - \frac{1}{n}\right)^{\frac{n}{n} km}$$

$$\simeq \left(\frac{1}{e}\right)^{\frac{km}{n}} = e^{-km/n}$$

using the fact that for large n $(1 - 1/n)^n \simeq \frac{1}{e}$

Define $p = e^{-km/n}$ and assume that A contains "exactly" $p \cdot n$ 0's

Consider $x_i \notin S$ and let

$l_j = h_j(x_i)$ index returned by h_j for x_i

Prob(x_i is erroneously deemed to be in S) =

$$\text{Prob}(A[\ell_1] = A[\ell_2] = \dots = A[\ell_k] = 1) =$$

$$\prod_{j=1}^{k} \text{Prob}(A[\ell_j] = 1) =$$

$$\prod_{j=1}^{k} (1 - \text{Prob}(A[\ell_j] = 0)) = (1 - p)^k$$

$$= (1 - e^{-km/n})^k$$

$\square$

Obs. Using standard calculus it can be proved that for fixed n, m the best choice for K is $K = \frac{n}{m} \ln 2$

(closest integer to $\frac{n}{m} \ln 2$)

Exercise

Consider a stream $\Sigma = x_1, x_2, \ldots, x_n$ of n measurements from sensors. Each measurement x_i is a pair (k_i, w_i), where k_i is the ID of a sensor and w_i is the value of the measurement (an integer). For a given sensor u occurring in Σ define

$$f_u = \sum_{(k_i, w_i) \in \Sigma \,:\, k_i = u} w_i,$$

i.e., the aggregate measurements taken by u.

1. Briefly describe a space-efficient unbiased estimator for $\sum_u (f_u)^2$, where the sum is over all sensors occurring in the stream.
2. What can you say about the unbiasedness of your estimator?

Exercise

Consider a Bloom filter built to assess membership for a set S of m elements. The Bloom filter consists of a n-bit array A and k hash functions $h_1, h_2, \ldots, h_k$, mutually independent and with values uniformly distributed in $[0, n-1]$. Assume that n is even, and that each hash function h_i is such that, for every $e \in S$, $h_i(e) \bmod n/2$ is uniformly distributed in $[0, n/2-1]$.

1. Show how to transform, in $O(n)$ time, the given Bloom filter into a new Bloom filter based on an $n/2$-bit array B, and describe how to assess membership with the new Bloom filter.
2. Compute the probability that a given cell $B[i]$ of the new array is 0.

References

- **[LRU14]** J. Leskovec, A. Rajaraman and J. Ullman. Mining Massive Datasets. Cambridge University Press, 2014. Chapter 4 (Sections 4.3-4.5) (pdf provided in Moodle)

- **[CGHJ12]** G. Cormode, M.N. Garofalakis, P.J. Haas, C. Jermaine. Synopses for Massive Data: Samples, Histograms, Wavelets, Sketches. Foundations and Trends in Databases 4(1-3): 1-294, 2012. Chapter 5 (Sections 5.1-5.3) (pdf provided in Moodle)