Mining Data Streams-Finding Distinct Element

Barna Saha

February 9, 2016

Counting Distinct Items

Given a stream of elements arriving from a universe, we want to count the number of distinct elements in the stream, either from the beginning of the stream, or from some known time in the past.

Let S be a multi-set of N integers. Each integer is in the range $[0, U]$ where U is some polynomial in N. The *distinct element counting problem* finds out exactly how many distinct elements are there in S.

Motivating Example: Unique Users of a Website

Web sites often gather statistics on how many unique users it has seen in each given month. The universal set is the set of logins for that site, and a stream element is generated each time a user logs in.

- Amazon: user logs in with their unique login name.
- Google: identifies users by IP addresses.

Motivating Example: RFID Counting

Radio-frequency identification (RFID) technology uses RFID tags and RFID readers (or simply called tags and readers) to monitor objects in physical world.

Many events (e.g., TechEd and Bonnaroo festival) distribute RFID wristbands to their visitors. RFID counting helps reveal the number of people around.

Motivating Example: DNA Motifs

Sequence motifs are short, recurring patterns in DNA that are presumed to have a biological function.

Number of distinct motifs indicate valuable biological information about the specific DNA sequence.

Applications to Networks

- How many packet flows between distinct pairs of (source, destination)?
- How many flows are losing packets (where packets in one side not equal to packets out)?
- Denial of service attacks signaled by large numbers of requests from spoofed IPs.

Counting distinct elements provide valuable statistics in these cases.

A Simple Solution

- Keep an array, $a[0, .., U]$, initially set to 0.
- Also keep a counter C initialized to 0.
- Every time an item i arrives, look at $a[i]$.
- If it is zero, increment C, and set $a[i] = 1$
- Return C as the number of distinct items
- Time: $O(1)$ per update and per query
- But space is $O(U)$.

The Flajolet-Martin Sketch

- Counting the number of distinct items is easy if the items can be stored in the main memory.
- Store them in an easily searchable data structure such as a hash table, or search tree, and while adding an element, check if it is added for the first time to adjust the counter.
- What happens if we do not have enough memory to store all the distinct items?–The Flajolet-Martin Sketch.

The Flajolet-Martin Sketch

The basic idea.

- Keep an array $a[1....\log U]$
- Use a hash function $f : \{1...U\} \rightarrow \{0....\log U\}$
- Compute $f(i)$ for every item in the stream, and set $a[f(i)] = 1$.
- Somehow extract from this the approximate number of distinct items.

Space requirement$=O(\log U) = O(\log N)$, assuming hash functions do not require too much of space.

What kind of hash functions to use?

Universal Hash Function Family

Hash functions are uniform over $[M]$

$$Pr_{h \leftarrow \mathcal{H}}[h(i) = k] = \frac{1}{M}$$

$$Pr_{h \leftarrow \mathcal{H}}[h(i) = h(j)] \leq \frac{1}{M}$$

(2-universal hash family)

A family of hash functions $\mathcal{H} = \{h | h : [U] \rightarrow [M]\}$ is called a pairwise independent family of hash functions or strongly 2-universal if for all $i \neq j \in [U]$ and any $k, l \in [M]$

$$Pr_{h \leftarrow \mathcal{H}}[h(i) = k \cap h(j) = l] = \frac{1}{M^2}$$

Universal Hash Functions

Construction.

- Let p be a prime in $[U, 2U]$. For any $a, b \in \{0, 1, 2, ..., p-1\}$, define
- $h_{a,b}(x) = [(ax + b) \ mod \ p] mod \ n$ to obtain a hash function mapping $[U]$ to $[0, .., n-1]$.
- Then the collection of functions $\mathcal{H} = \{h_{a,b} \mid a, b \in [0, p-1]\}$ is strongly 2-universal.

The Flajolet-Martin Sketch

- We want a strongly 2-universal hash function family mapping $[U] \to [\lceil \log U \rceil]$.
- Let $U = 2^w - 1$. So $\lceil \log U \rceil = w$. Each integer $k \in [0, 2^w - 1]$ can be represented with w bits. To construct a hash function $f : [U] \to [\lceil \log U \rceil]$, we first pick a 2-universal hash function $h : [U] \to [U]$, and then look at the the number of trailing 0s of the outcome of h, to construct the mapping f.
- Let z_k denote the number of trailing 0's in the binary form of the hash $h(k)$ of k, where h is chosen from a strongly 2-universal hash family mapping from U to U. Then $f(k) = z_k$.
  - If $w = 5$, and $h(k) = 6 = (00110)_2$, then $z_k = 1$.
- FM sketch is simply an integer Z defined as:

$$Z = \max_{k \in S} z_k$$

- **Our estimate is simply**

$$\hat{F} = 2^Z$$

The Flajolet-Martin Sketch

- Probability that $z_k \geq 1 = \frac{1}{2}$
- Probability that $z_k \geq 2 = \frac{1}{4}$
- Probability that $z_k \geq 3 = \frac{1}{8}$
- Probability that $z_k \geq 4 = \frac{1}{16}$

Each item falls in the same cell every time it is encountered, so it is as if only one of each distinct item arrives.

Suppose you have 32 distinct elements. Then roughly 16 will have the least significant bit (LSB) as 0. Out of them 8 will have 2 LSBs set to 0, 4 will have 3 LSBs set to 0, 2 will have 4 LSBs set to 0, and 1 will have 5 LSBs set to 0. Then $Z = 5$, $2^Z = 32$.

The Flajolet-Martin Sketch: Analysis

We will prove

Theorem

For any integer $c > 3$ the probability that $\frac{F}{c} \leq \hat{F} \leq cF$ is at least $1 - \frac{3}{c}$ where F is the true value of distinct elements and $\hat{F}$ is the estimate from FM-sketch.

The Flajolet-Martin Sketch: Analysis

Lemma

For any integer $r \in [0, w]$, $Pr[z_k \geq r] = \frac{1}{2^r}$.

Proof.

There are w bits, among them the least r bits must all be 0. There are 2^{w-r} such integers in $[0, 2^w - 1]$ that have at least r trailing 0s. Therefore,

$$Pr[z_k \geq r] = \frac{2^{w-r}}{2^w} = \frac{1}{2^r}$$

□

The Flajolet-Martin Sketch: Analysis

Let us fix a r. For each $k \in S$, define:

$$x_k(r) = \begin{cases} 1, & \text{if } z_k \geq r \\ 0, & \text{otherwise} \end{cases}$$

$$E[x_k(r)] = Prob(x_k(r) = 1) = \frac{1}{2^r}$$

$$Var[x_k(r)] = E[x_k(r)^2] - (E[x_k(r)])^2 = \frac{1}{2^r} - \frac{1}{2^{2r}} = \frac{1}{2^r}\left(1 - \frac{1}{2^r}\right)$$

Define

$$X(r) = \sum_{k \ distinct} x_k(r)$$

We must have $X(r) \geq 1$ for $r = 0, 1, 2, ..., Z$. and $X(r) = 0$ for $r = Z + 1, ..., w - 1$

The Flajolet-Martin Sketch: Analysis

Define

$$X(r) = \sum_{k \ \ distinct} x_k(r)$$

We must have $X(r) \geq 1$ for $r = 0, 1, 2, ..., Z$. and $X(r) = 0$ for $r = Z + 1, ..., w - 1$
Let

$$r_1 = \text{ the smallest } r \text{ such that } 2^r > cF$$

$$r_2 = \text{ the largest } r \text{ such that } 2^r < \frac{F}{c}$$

For the algorithm to be successful we want $r_2 < Z < r_1$, or $X(r_2 + 1) > 0$ and $X(r_1) = 0$.

The Flajolet-Martin Sketch: Analysis

We will show

Lemma
$Pr[X(r_1) \geq 1] < \frac{1}{c}$

Lemma
$Pr[X(r_2 + 1) = 0] \leq \frac{2}{c}$

Therefore, we will have

$$Pr[X(r_1) \geq 1 \ OR \ X(r_2 + 1) = 0] < \frac{3}{c}$$

Thus the algorithm is successful with probability at least $1 - \frac{3}{c}$.

Markov Inequality

Theorem 3 (Markov Bound). *For any positive random variable X, and for any $t > 0$*

$$\Pr(X \geq t) \leq \frac{E[x]}{t} \tag{1}$$

Proof.

$$\begin{aligned} E[x] &= \sum_x x \cdot \Pr(X = x) \\ &= \sum_{x<t} x \cdot \Pr(X = x) + \sum_{x \geq t} x \cdot \Pr(X = x) \\ &\geq 0 + t \cdot \sum_{x \geq t} \Pr(X = x) \\ &= t \cdot P(X \geq t) \end{aligned}$$

□

Chebyshev Inequality

Theorem 4 (Chebyshev Inequality). *For any random variable X and for any $t > 0$*

$$\Pr(|X - E[x]| \geq t) \leq \frac{Var(x)}{t^2} \tag{2}$$

Proof.

$$\begin{aligned} & \Pr(|X - E[x]| \geq t) \\ = \; & Pr([X - E[x]]^2 \geq t^2) \\ \leq \; & \frac{E\left[(X - E[x])^2\right]}{t^2} \\ = \; & \frac{Var(X)}{t^2} \end{aligned}$$

□

The Flajolet-Martin Sketch: Analysis

Lemma
$Pr[X(r_1) \geq 1] < \frac{1}{c}$

Proof.

$$
\begin{aligned}
Pr[X(r_1) \geq 1] &\leq E[X(r_1)] && \text{by Markov Inequality} \\
&= \sum_{k \ distinct} E[x_k(r_1)] && \text{by linearity of expectation} \\
&= \frac{F}{2^{r_1}} < \frac{1}{c}
\end{aligned}
$$

□

The Flajolet-Martin Sketch: Analysis

Lemma

$Pr[X(r_2+1)=0] \leq \frac{2}{c}$

Proof.

$$Pr[X(r_2+1)=0] \leq Pr[|X(r_2+1) - E[X(r_2+1)| \geq E[X(r_2+1)]$$

$$= \frac{Var(\sum_{k \text{ distinct}} x_k(r_2+1))}{(\sum_{k \text{ distinct}} E[x_k(r_2+1)])^2}$$

$$= \frac{\sum_{k \text{ distinct}} Var(x_k(r_2+1))}{(\sum_{k \text{ distinct}} E[x_k(r_2+1)])^2} \quad \text{by 2-wise independence}$$

$$\leq \frac{F}{2^{(r_2+1)}} \Big/ \frac{F^2}{2^{2(r_2+1)}} = \frac{2^{r_2+1}}{F} \leq \frac{2}{c}$$

□

The Chernoff Bound

Theorem 5 (The Chernoff Bound). *Let $X_1, X_2 ... X_n$ be n independent Bernoulli random variables with $\Pr(X_i = 1) = p_i$. Let $X = \sum X_i$. Hence,*

$$E[X] = E\left[\sum X_i\right] = \sum E[X_i] = \sum \Pr(X_i = 1) = \sum p_i = \mu (say).$$

Then the Chernoff Bound says for any $\epsilon > 0$

$$\begin{aligned} \Pr(X > (1+\epsilon)\mu) &\leq \left(\frac{e^{\epsilon}}{(1+\epsilon)^{\epsilon}}\right)^{\mu} \text{ and} \\ \Pr(X < (1-\epsilon)\mu) &\leq \left(\frac{e^{-\epsilon}}{(1-\epsilon)^{1-\epsilon}}\right)^{\mu} \end{aligned}$$

When $0 < \epsilon < 1$ the above expression can be further simplified to

$$\begin{aligned} \Pr(X > (1+\epsilon)\mu) &\leq e^{\frac{-\mu\epsilon^2}{3}} \text{ and} \\ \Pr(X < (1-\epsilon)\mu) &\leq e^{\frac{-\mu\epsilon^2}{2}} \end{aligned}$$

Hence

$$\Pr(|X - \mu| > \epsilon\mu) \leq 2e^{\frac{-\mu\epsilon^2}{3}}$$

Java Implementation of FM-Sketch

https://github.com/rbhide0/Columbus/blob/master/src/main/java/rbhide0/streaming/algorithm/FlajoletMartin.java

Results using the above implementation.

- Wikipedia article on "United States Constitution" had 3978 unique words. When run ten times, Flajolet-Martin algorithm reported values of 4902, 4202, 4202, 4044, 4367, 3602, 4367, 4202, 4202 and 3891 for an average of 4198. As can be seen, the average is about right, but the deviation is between -400 to 1000.
- Wikipedia article on "George Washington" had 3252 unique words. When run ten times, the reported values were 4044, 3466, 3466, 3466, 3744, 3209, 3335, 3209, 3891 and 3088, for an average of 3492.

Play with this implementation, and let me know what you counted!

Reference: http://ravi-bhide.blogspot.com/2011/04/flajolet-martin-algorithm.html