L12 -- Heavy Hitters in Streams
[Jeff Phillips - Utah - Data Mining]

Streaming Algorithms

Stream : \( A = \langle a_1, a_2, \ldots, a_m \rangle \)
\( a_i \in [n] \) size \( \log n \)
Compute \( f(A) \) in \( \text{poly}(\log m, \log n) \) space
"one pass"

Let \( f_j = |\{a_i \in A \mid a_i = j\}| \)
\( F_1 = \sum_j f_j = m \) == total count

Goal: Find all \( j \) s.t. \( f_j > \phi m \)
\( \phi = 1/k = \varepsilon \)

\( f_j - \varepsilon m \leq \hat{f}_j \leq f_j \) Misra-Greis [1985]
\( f_j \leq \hat{f}_j \leq f_j + \varepsilon m \) Count-Min [Cormode + Muthukrishnan '05]

-------------------------

FP-MAJORITY: if some \( f_j > m/2 \), output \( j \)
else, output anything

How good w/ \( O(\log m + \log n) \) (one counter \( c \) + one location \( l \))? ...

####################################
c = 0, \( l = X \)
for \( (a_i \in A) \)
    if \( (a_i = l) \) \( c += 1 \)
    else \( c -= 1 \)
    if \( (c <= 0) \) \( c = 1, \ l = a_i \)
return \( l \)
####################################

Analysis: if \( f_j > m/2 \), then
if \( (l \neq j) \) then \( c \) decremented at most \( < m/2 \) times, but \( c > m/2 \)
if \( (l = j) \) can be decremented \( < m/2 \), but is incremented \( > m/2 \)
if \( f_j < m/2 \) for all \( j \), then any answer ok.

-----------------------

k-FREQUENCY-ESTIMATION: Build data structure \( S \).
For any $j$ in $[n]$, $\hat{f}_j = S(j)$ s.t.
\[ f_j - \frac{m}{k} \leq \hat{f}_j \leq f_j \]

aka eps-approximate phi-HEAVY-HITTERS:
- Return all $f_j$ s.t. $f_j > \phi m$
- Return no $f_j$ s.t. $f_j < \phi m - \epsilon m$
- (any $f_j$ s.t. $\phi m - \epsilon m < f_j < \phi m$ is ok)

-------------------------

Misra-Gries Algorithm [Misra-Gries '82]

Solves $k$-FREQUENCY-ESTIMATION in $O(k(\log m + \log n))$ space.

Let $C$ be array of $k$ counters $C[1], C[2], \ldots, C[k]$
Let $L$ be array of $k$ locations $L[1], L[2], \ldots, L[k]$

############################
Set all $C = 0$
Set all $L = X$

for $(a_i \in A)$
  if $(a_i \in L)$  \textit{<at index $j$>}
    $C[j] += 1$
  else  \textit{<a_i !in L>}
    if $(|L| < k)$
      $C[j] = 1$
      $L[j] = a_i$
    else
      $C[j] -= 1 \text{forall } j \in [k]$

for $(j \in [k])$
  if $(C[j] <= 0)$ set $L[j] = X$

############################
On query $q$ in $[n]$
  if $(q \in L \{L[j]=q\})$ return $\hat{f}_q = C[j]$
  else return $\hat{f}_q = 0$

############################

Analysis

A counter $C[j]$ representing $L[j] = q$ is only incremented if $a_i = q$

\[ \hat{f}_q \leq f_q \]
If a counter $C[j]$ representing $L[j] = q$ is decremented, then $k-1$ other counters are also decremented. This happens at most $m/k$ times.

A counter $C[j]$ representing $L[j] = q$ is decremented at most $m/k$ times.

$$f_q - m/k \leq \hat{f}_q$$

How do we get an additive $\varepsilon$-approximate FREQUENCY-ESTIMATION?

i.e. return $\hat{f}_q$ s.t.

$$|f_q - \hat{f}_q| \leq \varepsilon m$$

Set $k = 2/\varepsilon$, return $C[j] + (m/k)/2$

Space $O((1/\varepsilon) \log m + \log n))$

Also:

$\varepsilon$-approximate phi-HEAVY-HITTERS for any $\phi > m \varepsilon$ in space $O((1/\varepsilon) \log m + \log n))$

---------------------------

COUNT MIN Sketch
---------------------------

$t$ independent hash functions $\{h_1, ..., h_t\}$
Each $h_i : [n] \to [k]$

2-d array of counters:
$h_1 \to [C_{1,1}] [C_{1,2}] ... [C_{1,k}]$
$h_2 \to [C_{2,1}] [C_{2,2}] ... [C_{2,k}]$

... ... ... ... ...
$h_t \to [C_{t,1}] [C_{t,2}] ... [C_{t,k}]$

For each $a \in A$ -> increment $C_{i,h_i(a)}$ for $i \in [t]$

$\hat{f}_a = \min_{i \in [t]} C_{i,h_i(a)}$

Set $t = \log(1/\delta)$
Set $k = 2/\varepsilon$

---------------------------

Clearly $f_a \leq \hat{f}_a$

$\hat{f}_a \leq f_a + W$. What is $W$?
One hash function \( h_i \).
Adds to \( W \) when there is a collision \( h_i(a) = h_i(j) \).  \( \text{wp} \ 1/k \)

random variable \( Y_{i,j} \)
\[ Y_{i,j} = \{ f_j \ \text{wp} \ 1/k, \ 0 \ \text{wp} \ 1-1/k \} \]
\[ E[Y_{i,j}] = f_j/k \]

random variable \( X_{i} = \sum_{j \in [n], j\neq a} Y_{i,j} \)
\[ E[X_{i}] = E[\sum_{j} Y_{i,j}] = \sum_{j} f_j/k = F_1/k = \epsilon * F_1/2 \]

Markov Inequality
\( X \) a rv and \( \alpha > 0 \)
\[ \Pr[|X| \geq \alpha] \leq \frac{E[|X|]}{\alpha} \]

\( X_{i} > 0 \) so \( |X_{i}| = X_{i} \)
setting \( \alpha = \epsilon F_1 \) then
\[ E[|X|]/\alpha = (\epsilon*F_1/2)/(\epsilon F_1) = 1/2 \]
\[ \Pr[X_{i} \geq \epsilon F_1] \leq 1/2 \]

-----
Now for \( t \) *independent* hash functions:

\[ \Pr[\hat{f}_a - f_a \geq \epsilon F_1] = \Pr[\min_i X_{i} \geq \epsilon F_1] \]
\[ = \Pr[\forall_{i \in [t]} (X_{i} \geq \epsilon F_1)] \]
\[ = \prod_{i \in [t]} \Pr[X_{i} \geq \epsilon F_1] \]
\[ \leq 1/2^t \]
\[ = \delta (\text{since} \ t = \log(1/\delta)) \]

Hence:
\[ f_a \leq \hat{f}_a \leq f_a + \epsilon F_1 \]
- first inequality always holds
- second inequality holds \( \text{wp} > 1-\delta \)

-----
Space:
each of \( k*t \) counters requires \( \log m \) space
\( O(k*t*\log m) \)
Store \( t \) hash functions: \( \log n \) each
\( O((k \log m + \log n)*t) = O((1/\epsilon \log m + \log n) \log (1/\delta)) \)

--------------------------
turnstile model:  add or subtract (as long as is there)