Lecture 4: CountSketch High Frequencies
Plan

• Scriber?

• Plan:
  – CountMin/CountSketch (continuing from last time)
  – High frequency moments via Precision Sampling
Part 1: CountMin/CountSketch

- Let \( f_i \) be frequency of \( i \)

- Last lecture:
  - \( 2^{\text{nd}} \) moment: \( \sum_i f_i^2 \)
    - Tug of War
  - \( \text{Max}: \) heavy hitter
    - CountMin

<table>
<thead>
<tr>
<th>IP</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>9</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>...</td>
<td>0</td>
</tr>
<tr>
<td>( n )</td>
<td>1</td>
</tr>
</tbody>
</table>
CountMin: overall

- **Heavy hitters:** $\frac{\hat{f}_i}{\sum f_j} \geq \phi$
  - If $\frac{f_i}{\sum f_j} \leq \phi (1 - \epsilon)$, not reported
  - If $\frac{f_i}{\sum f_j} \geq \phi (1 + \epsilon)$, reported as heavy hitter

- **Space:** $O \left( \frac{\log n}{\epsilon \phi} \right)$ cells

---

**Algorithm CountMin:**

Initialize($L$, $w$):
- array $S[L][w]$
  - $L$ hash functions $h_1 \ldots h_L$, into $\{1, \ldots, w\}$

Process(int $i$):
- for($j=0$; $j<L$; $j++$)
  - $S[j][ h_j(i) ] += 1$;

Estimator:
- foreach $i$ in PossibleIP {
  - $\hat{f}_i = \min_j (S[j][h_j(i)])$;
}
• Can improve time; space degrades to $O\left(\frac{\log^2 n}{\epsilon \phi}\right)$

• **Idea:** dyadic intervals
  – Each level: one CountMin sketch on the virtual stream
  – Find heavy hitters by following down the tree the heavy hitters

\[
\begin{align*}
\sum_{i=1}^{n/2} f_i &= \sum_{i=\frac{n}{2}+1}^{n} f_i \\
\sum_{i=1}^{\frac{n}{2}} f_i &= \sum_{i=\frac{n}{2}+1}^{n} f_i \\
\sum_{i=1}^{n} f_i &= \sum_{i=\frac{n}{2}+1}^{n} f_i \\
\end{align*}
\]
CountMin: linearity

• Is CountMin linear?
  – CountMin($f' + f''$) from CountMin($f'$) and CountMin($f''$)?
  – Just sum the two!
    • sum the 2 arrays, assuming we use the same hash function $h_j$

• Used a lot in practice

https://sites.google.com/site/countminsketch/
**CountSketch**

- How about $f = f' - f''$?
  - Or general streaming
  - “Heavy hitter”:
    - if $|f_i| \geq \phi \sum_j |f_j| = \phi \cdot ||f||_1$
  - “min” is an issue
  - But median is still ok

- Ideas to improve it further?
  - Use Tug of War $r$ in each bucket => CountSketch
  - Better in certain sense (cancelations in a cell)

---

Algorithm CountSketch:

Initialize(L, w):

- array $S[L][w]$
- L hash func’s $h_1 \ldots h_L$, into [w]
- L hash func’s $r_1 \ldots r_L$, into {±1}

Process(int i, real $\delta_i$):

- for(j=0; j<L; j++)
  - $S[j][h_j(i)] += r_j(i) \cdot \delta_i$;

Estimator:

- foreach i in PossibleIP {
  - $\hat{f}_i = median_j(S[j][h_j(i)])$;
CountSketch ⇒ Compressed Sensing

• Sparse approximations:
  – \( f \in \mathbb{R}^n \)
  – \( k \)-sparse approximation \( f^* \):
    • \( \min ||f^* - f|| \)
    • Solution: \( f^* = \) the \( k \) heaviest elements of \( f \)

• Compressed Sensing:
  [Candes-Romberg-Tao’04, Donoho’04]
  – Want to acquire signal \( f \)
  – Acquisition: linear measurements (sketch) \( S(f) = Sf \)
  – Goal: recover \( k \)-sparse approximation \( \hat{f} \) from \( Sf \)
  – Error guarantee:
    \[ ||\hat{f} - f|| \leq \min_{k\text{-sparse } f^*} ||f^* - f|| \]
  – Theorem: need only \( O(k \cdot \log n) \)-size sketch!
Signal Acquisition for CS

• Single pixel camera
  [Takhar-Laska-Waskin-Duarte-Baron-Sarvotham-Kelly-Baraniuk’06]
  • One linear measurement = one row of $S$

  ![Diagram of signal acquisition](http://dsp.rice.edu/sites/dsp.rice.edu/files/cs/cscam-SPIEJan06.pdf)

• CountSketch: a version of Compr Sensing
  – Set $\phi = 1/2k$
  – $\hat{f}$: take all the heavy hitters (or $k$ largest)
  – Space: $O(k \log n)$
Back to Moments

- General moments:
  - $p^{th}$ moment: $\sum_i f_i^p$
    - normalized: $(\sum_i f_i^p)^{1/p}$
  - $p = 2$: $\sum f_i^2$
    - $O(\log n)$ via Tug of War (Lec. 3)
  - $p = 0$: count # distinct!
    - $O(\log n)$ [Flajolet-Martin] from Lec. 2
  - $p = 1$: $\sum |f_i|$
    - $O(\log n)$: will see later (for all $p \in (0,2)$)
  - $p = \infty$ (normalized): $\max_i f_i$
    - Impossible to approximate, but can heavy hitters (Lec. 3)
  - Remains: $2 < p < \infty$?
    - Space: $\Theta\left(n^{1-\frac{2}{p}} \log^2 n\right) \Rightarrow$ Precision Sampling (next)
A task: estimate sum

- Given: \( n \) quantities \( a_1, a_2, \ldots, a_n \) in the range \([0,1]\)
- Goal: estimate \( S = a_1 + a_2 + \cdots + a_n \) “cheaply”

- Standard sampling: pick random set \( J = \{j_1, \ldots, j_m\} \) of size \( m \)
  - Estimator: \( \tilde{S} = \frac{n}{m} \cdot (a_{j_1} + a_{j_2} + \cdots + a_{j_m}) \)
- Chebyshev bound: with 90% success probability
  \[ S - O(n/m) < \tilde{S} < S + O(n/m) \]
- For constant additive error, need \( m = \Omega(n) \)

Compute an estimate \( \tilde{S} \) from \( a_1, a_3 \)
• Alternative “access” to $a_i$’s:
  – For each term $a_i$, we get a (rough) estimate $\tilde{a}_i$
  – up to some precision $u_i$, chosen in advance:
    $|a_i - \tilde{a}_i| < u_i$

• Challenge: achieve good trade-off between
  – quality of approximation to $S$
  – use only weak precisions $u_i$ (minimize “cost” of estimating $\tilde{a}$)

Compute an estimate $\tilde{S}$ from $\tilde{a}_1, \tilde{a}_2, \tilde{a}_3, \tilde{a}_4$
Formalization

Sum Estimator

1. fix precisions $u_i$

3. given $\tilde{a}_1, \tilde{a}_2, ... \tilde{a}_n$, output $\tilde{S}$ s.t.
\[ |\sum_i a_i - \gamma \tilde{S}| < 1 \text{ (for } \gamma \approx 1) \]

- What is cost?
  - Here, average cost = $1/n \cdot \sum 1/u_i$
  - to achieve precision $u_i$, use $1/u_i$ “resources”: e.g., if $a_i$ is itself a sum $a_i = \sum_j a_{ij}$ computed by subsampling, then one needs $\Theta(1/u_i)$ samples

- For example, can choose all $u_i = 1/n$
  - Average cost $\approx n$

Adversary

1. fix $a_1, a_2, ... a_n$

2. fix $\tilde{a}_1, \tilde{a}_2, ... \tilde{a}_n$ s.t. $|a_i - \tilde{a}_i| < u_i$
Precision Sampling Lemma

• Goal: estimate $S = \sum a_i$ from $\{\tilde{a}_i\}$ satisfying $|a_i - \tilde{a}_i| < u_i$.

• **Precision Sampling Lemma:** can get, with 90% success:
  - $O(1)$ additive error and $1.5$ multiplicative error:
    $S/1.5 - O(1) < \tilde{S} < 1.5 \cdot S + O(1)$
  - with average cost equal to $O(\log n)$

• Example: distinguish $\Sigma a_i = 3$ vs $\Sigma a_i = 0$
  - Consider two extreme cases:
    • if three $a_i = 1$: enough to have crude approx for all ($u_i = 0.1$)
    • if all $a_i = 3/n$: only few with good approx $u_i = 1/n$, and the rest with $u_i = 1$
• **Precision Sampling Lemma:** can get, with 90% success:
  - $O(1)$ additive error and $1.5$ multiplicative error:
    $$S/1.5 - O(1) < \tilde{S} < 1.5 \cdot S + O(1)$$
  - with average cost equal to $O(\log n)$

• **Algorithm:**
  - Choose each $u_i \in \text{Exp}(1)$ i.i.d.
  - Estimator: $\tilde{S} = \max_i \frac{\tilde{a_i}}{u_i}$.

• **Proof of correctness:**
  - **Claim 1:** $\max a_i/u_i \sim \sum a_i/\text{Exp}(1)$
    - Hence, $\max \tilde{a_i}/u_i = \frac{\sum a_i}{\text{Exp}(1)} \pm 1$
  - Claim 2: Avg cost $= O(\log n)$
\(p\)-moments via Prec. Sampling

**Theorem:** linear sketch for \(p\)-moment with \(O(1)\) approximation, and \(O(n^{1-2/p} \log^{O(1)} n)\) space (with 90% success probability).

**Sketch:**
- Pick random \(r_i \in \{\pm 1\}\), and \(u_i \sim \text{Exp}(1)\)
- let \(y_i = f_i \cdot r_i / u_i^{1/p}\)
- Hash into a hash table \(S\),
  \[w = O(n^{1-2/p} \log^{O(1)} n)\] cells

**Estimator:**
- \(\max_j |S[j]|^p\)

**Linear**
\[
f = f_1 f_2 f_3 f_4 f_5 f_6
\]
\[
S = y_1 + y_3\quad y_4\quad y_2 + y_5 + y_6
\]

\(u \sim e^{-u}\)
Correctness of estimation

- **Theorem:** \(\max_j |S[j]|^p\) is \(O(1)\) approximation with 90% probability, with 
  \(w = O(n^{1-2/p} \log^{O(1)} n)\) cells

- **Proof:**
  - Use Precision Sampling Lem.
  - \(a_i = |f_i|^p\)
    - \(\sum a_i = \sum |f_i|^p = F_p\)
  - \(\tilde{a}_i = |S[h(i)]|^p\)
  - Need to show \(|a_i - \tilde{a}_i|\) small
    - more precisely: \(\left|\frac{\tilde{a}_i}{u_i} - \frac{a_i}{u_i}\right| \leq \varepsilon F_p\)

**Algorithm PrecisionSamplingFp:**

Initialize(w):
- array \(S[w]\)
- hash func \(h\), into \([w]\)
- hash func \(r\), into \(\{\pm 1\}\)
- reals \(u_i\), from \(Exp\) distribution

Process(vector \(f \in \mathbb{R}^n\)):
- for \(i=0; i<n; i++\)
  - \(S[h(i)] += f_i \cdot \frac{r_i}{u_i^{1/p}};\)

Estimator:
- \(\max_j |S(j)|^p\)
Correctness 2

- **Claim:**  
  \[ |S[h(i)]^p - f_i^p / u_i| \leq O(\epsilon F_p) \]
- Consider cell \( z = h(i) \)
  - \( S[z] = \frac{f_i}{u_i^{1/p}} + C \)
- How much chaff \( C \) is there?
  - \( C = \sum_{j \neq i^*} y_j \cdot \chi[h(j) = z] \)
  - \( E[C^2] = \cdots \leq ||y||^2 / w \)
  - What is \( ||y||^2 \)?
    - \( E_u ||y||^2 \leq ||f||^2 \cdot E \left[ \frac{1}{u_i^{2/p}} \right] = ||f||^2 \cdot O(\log n) \)
    - \( ||f||^2 \leq n^{1-2/p} ||f||_p^2 \)
    - By Markov’s: \( C^2 \leq ||f||_p^2 \cdot n^{1-2/p} \cdot O(\log n) / w \) with probability >90%
- Set \( w = \frac{1}{\epsilon^2/p} n^{1-2/p} \cdot O(\log n) \), then
  - \( |C|^p \leq ||f||_p^p \cdot \epsilon = \epsilon F_p \)

\[ \begin{align*}
  y_1 & = f_i \cdot r_i / u_i^{1/p} \\
  u_i & \text{ exponential r.v.}
\end{align*} \]
Correctness (final)

• **Claim:** \( |S[h(i)]^p - f_i^p / u_i| \leq O(\epsilon F_p) \)

• \( S[h(i)]^p = \left( \frac{f_i}{u_i^{1/p}} + C \right)^p \)
  – where \( C = \sum_{j \neq i^*} y_j \cdot \chi[h(j) = h(i)] \)

• **Proved:**
  – \( E[C^2] \leq ||y||^2 / w \)
  – this implies \( C^p \leq \epsilon F_p \) with 90% for fixed \( i \)
  – But need for all \( i \)!

• **Want:** \( C^2 \leq \beta ||y||^2 / w \) with high probability for some smallish \( \beta \)
  – Can indeed prove for \( \beta = O(\log^2 n) \) with strong concentration inequality (Bernstein).
Recap

• CountSketch:
  – Linear sketch for general streaming

• $p$-moment for $p > 2$
  – Via Precision Sampling
    • Estimate of sum from poor estimates
  – Sketch: Exp variables + CountSketch