COMS 4995-3: Advanced Algorithms — Feb 6, 2017

Lecture 6 – Sketching, Frequency moments

Instructor: *Alex Andoni* — Scribes: *Saurabh Bondarde, Ying Sheng*

1 Last Time

Last time we covered the Count Min Sketch Algorithm and concluded with the following claims.
Given $\epsilon, \phi > 0$, the sketch for finding ϕ Heavy hitter's problem

- occupies $O(\frac{1}{\epsilon\phi} \log n)$ space (or words)
- takes $O(\log n)$ time per update
- takes $O(n \log n)$ time to find the actual heavy hitter
  - report all i such that $\widehat{f_i} \geq \phi \sum f_i$
  - do not report any i such that $\widehat{f_i} < (1-\epsilon)\phi \sum f_i$

2 Introduction

Today's lecture is about extending the Count Min algorithm for more than one routers. Up till now, we only analyzed streams going through one router. The approach to get aggregate statistics of streams over multiple routers will be discussed today.

3 Count Min Linearity

Imagine we have a stream of size n, the stream $\{x_1, x_2, ..., x_n\}$, with its corresponding sketch S_1, and another stream of size m, $\{y_1, y_2, ..., y_m\}$, with sketch S_2. Note that each x_i, $y_i \in [n]$. Now suppose we want the statistics on the union of the two streams, for example if we want the ϕ-heavy hitters of $f + g$ where f and g are the corresponding frequency estimates. (The + here indicates aggregate or union over the two streams).

Approach: Composition! If f and g use the same hash function, then the composition of both the streams is the sketch '$f + g$'.

As we have seen earlier that with a single stream, the algorithm to identify heavy hitters requires enumeration over all possible i. This step can take a lot of time especially if you consider aggregate statistics of compositions. Can we reduce this time? Yes!

4 Optimized algorithm

Theorem 1. *Without enumerating for all i, we can get the same result as the Count Min Algorithm using*

– $O(\frac{log^2 n}{\phi})$ *for finding the heavy hitters*

– $O(\frac{log^2}{\epsilon\phi})$ *space (words)*

Proof. The idea here is to use dyadic intervals i.e. find the heavy hitters using binary search.

□

Definition 2. *Dyadic Interval: A dyadic interval is a bounded interval whose endpoints are $\frac{j}{2^n}$ and $\frac{j+1}{2^n}$, where j and n are integers.*

Imagine a binary search tree structure for elements from 1 to n. The dyadic intervals from the root to the lower levels are as follows:
Root $\rightarrow [1, n]$
Level 1 $\rightarrow [1, \frac{n}{2}], [\frac{n}{2}+1, n]$
Level 2 $\rightarrow [1, \frac{n}{4}], [\frac{n}{4}+1, \frac{n}{2}], [\frac{n}{2}+1, \frac{3n}{4}], [\frac{3n}{4}+1, n]$
and so on ...

For each level $j \in \{0, ..., \log n\}$ (Note that height of the tree is $\log n$),

– it has 2^j items which correspond to intervals of length $\frac{n}{2^j}$

– think of these items as a stream over these intervals with frequencies $f^j \in \mathbb{N}^{2^j}$. f_I^j is the sum of the frequencies of items $i \in I$ for the particular interval.

– construct count min sketch on these 2^j virtual items. Example, at level 2, every item will fall in frequency of 1st interval $(1, \frac{n}{2})$ or 2nd interval $(\frac{n}{2}, n)$

Thus, $\sum_I f_I^j = \sum_{i=1}^n f_i = m$.

Space requirements: $O(log(n)).O(\frac{log(n)}{\epsilon\phi}) = O(\frac{log^2(n)}{\epsilon\phi})$
But question is how do we find heavy hitters faster? We can zoom in on them using the higher levels.

Claim 3. *If item i is ϕ-heavy $\Rightarrow$ parent is always ϕ-heavy.*
In other words, if interval I_1 is ϕ-heavy, then parent $(I = I_1 \bigcup I_2)$ is also ϕ-heavy.

$$f_I^{j-1} = f_{I_1}^j + f_{I_2}^j$$

Note: It is possible that the parent is ϕ-heavy even if none of its children being ϕ-heavy.

Final Algorithm

– start from root

– descend into children which are heavy (stop at any point where it is not ϕ-heavy)

Claim 4. *Any level can have* $\leq \frac{1}{\phi}$ *heavy items.*

Proof. We know $f_I \geq \phi \sum f_I$ This means there can be only $\frac{1}{\phi}$ □

Total time: $O(log(n)).O(\frac{1}{\phi}).O(log(n)) = O(\frac{log^2(n)}{\phi})$

Note: One open issue is that there is a probability of success associated. Here, Pr[Failure] $\leq \frac{1}{n}$.

5 Frequency Moments

Definition 5. *F_p denotes the p-th($p > 0$) moment of function f, which equals:*

$$F_p \triangleq \sum f_i^p$$

Next, we want to estimate $\widehat{F_2} = \sum f_i^2$.

Motivation: Why do we calculate this 2nd moment?
- application in databases to estimate size of database-joins
- connected with dimension reduction
- used for notion of error and calculating squares of error

Tug-of-War(ToW) Algorithm

Suppose the items in stream are in $[n]$

- Pick $\sigma_i \in \{\pm 1\}, i = 1, ..., n$ randomly
- Sketching to get: $z = \sum_{i=1}^{n} \sigma_i f_i$
  - On item $x_c = i$, update $z_{new} = z + \sigma_i$
- Output estimator: z^2

Actually, we can treat $\sigma : [n] \to \{-1, +1\}$ as random hash function. Let

$$z_+ = \sum_{i:\sigma_i=+1} f_i, z_- = \sum_{i:\sigma_i=-1} f_i$$

The estimator is equals to $(z_+ - z_-)^2$.

Next, we'll show how good is the estimator.

Claim 6. $\mathbb{E}_\sigma[z^2] = F_2$

Proof.

$$\begin{aligned}\mathbb{E}[z^2] =&\mathbb{E}[(\sum_{i=1}^{n}\sigma_i f_i)^2]\\ =&\mathbb{E}[\sum_{i=1}^{n}\sigma_i^2 f_i^2+\sum_{i\neq j}\sigma_i f_i\sigma_j f_j]\\ =&\sum_{i=1}^{n}f_i^2+\sum_{i\neq j}f_i f_j\mathbb{E}[\sigma_i\sigma_j]\\ =&\sum_{i=1}^{n}f_i^2\\ =&F_2\end{aligned}$$

□

Claim 7. $Var[z^2]=O(F_2^2)$

Proof.

$$\begin{aligned}Var[z^2] =&\mathbb{E}[z^4]-\mathbb{E}[z^2]^2\\ =&\sum_{i=1}^{n}f_i^4+\sum_{i\neq j}f_i^2 f_j^2\cdot 3\\ \leq&(\sum_{i=1}^{n}f_i^2)^2+3(\sum_{i=1}^{n}f_i^2)^2\\ \leq&4F_2^2\end{aligned}$$

□

Then, by Chebyshev's inequality, there is:

$$Pr\left[z^2 \text{ within } F_2\pm 3\sqrt{4F_2^2}\right]=Pr\left[z^2 \text{ within } F_2\pm 6F_2\right]\geq\frac{2}{3}$$

We can repeat the ToW algorithm multiple times to improve the accuracy.

ToW+: repeat ToW k times, and take the average of the estimators.

- Suppose z_j is the ToW sketch for independent $\{\sigma_{ji}\}, i=1,...,n$
- Output Estimator $z=\frac{1}{k}(\sum_{i=1}^{k}z_i^2)$

We can calculate the expectation and variance of the estimator as follows:

$$\mathbb{E}[z]=\mathbb{E}[\frac{1}{k}\sum_{j}z_j^2]=\frac{1}{k}\sum_{j=1}^{k}F_2=F_2$$

$$Var[z] = Var[\frac{1}{k}\sum_j z_j^2] = \frac{1}{k}Var[z_1^2] \leq \frac{4F_2^2}{k}$$

Again, by Chebyshev's inequality, there is:

$$Pr\left[z \text{ within } F_2 \pm \frac{\epsilon}{\sqrt{k}}F_2\right] \geq \frac{2}{3}$$

Then, we can set $k = O(\frac{1}{\epsilon^c})$ to get $(1+\epsilon)$ approximation.

6 Next time

Theorem 8. ***(Central Limit Theorem)***. *Let $X_1, X_2, ..., X_n$ be a random sample from a distribution(any distribution) with (finite) mean μ and (finite) variance σ^2. Let $Y = \frac{1}{k}\sum_{i=1}^{k} X_i$, and k sufficiently large. Then,*

- *The sample mean Y follows an approximate normal distribution.*
- $\mathbb{E}[Y] = \mu$
- $Var[Y] = \frac{\sigma^2}{n}$

Or say,

$$Y \xrightarrow{distribution} \mathcal{N}(\mu, \frac{\sigma^2}{n}) \text{ as } n \to \infty$$