CS388R: Randomized Algorithms Fall 2015

Lecture 18 — Nov 09, 2015

Prof. Eric Price *Scribe: Jin Zhang, Rishab Goyal*

1 Overview

In the previous lecture, we discussed

1. The minimum sample size to estimate fraction p of an unknown set, conforming certain property, within absolute relative error ϵ and at most δ failure probability.
2. How to find median of a list of elements with time complexity $\frac{3}{2}n + o(n)$.
3. (Briefly) the streaming algorithm to find α-heavy hitters (HH_α) after seeing $v_1, v_2, \ldots, v_m$ items, deterministically.

In this lecture, we revisit the count-min sketch problem, and analyze it in detail. First, we recall the *turnstile* model, and introduce some notation that we will follow throughout the scribe.

2 Introduction

Streaming algorithms are (informally) defined as algorithms that work on (massive) streams of data given only limited space and computational resources. The data stream model we will be interested in is called *turnstile* model. In turnstile model, each update is of the form (v_i, t_i), where v_i is a data point/ item, and $t_i \in \mathbb{Z}$ is a count. Throughout the scribe, we will denote by x_u, the number of times item u appeared in the stream. Concretely, after each update (v_i, t_i), $x_{v_i} \mathrel{+}= t_i$. Also, we define $\mathbf{x} = (x_{u_1}, \ldots, x_{u_n})$, $n = ||\mathbf{x}||_0$ and $m = ||\mathbf{x}||_1$, where u_i's are all distinct items.

In the standard turnstile model, x_{v_i} could be negative, however here we consider *strict* turnstile model, where x_{v_i} is non-negative for all items v_i at all time steps. We would like to find the set of α-heavy hitters (HH_α), where $\text{HH}_\alpha := \{u \mid x_u \geq \alpha m\}$, where m, n are same as defined before. Instead of solving perfectly, we solve an approximate version of the heavy hitters problem in the strict turnstile model. The $(\epsilon, \delta) - \text{HH}_\alpha$ is defined as finding a set S subject to the constraint that $S \supseteq \text{HH}_\alpha$ and $\forall u \in S$, $x_u \geq (1-\epsilon)\alpha m$, with failure probability at most δ. We could even find a set S such that $\forall u \in S$, $x_u \in [(1-\epsilon)\alpha m, (1+\epsilon)\alpha m]$ using the same idea.

In the next section, we describe count-min sketch, and show how to use it to solve $(\epsilon, \delta) - \text{HH}_\alpha$.

3 Count-min sketch

Count-min sketch could be considered as a counting Bloom filter using pairwise independent hash functions. In count-min sketch, we first make a sketch $Y \in \mathbb{Z}^{r \times c}$, such that $r = O\left(\log \frac{1}{\delta}\right)$,

$c = O\left(\frac{1}{\alpha}\right)$. For each row i, we use a different pairwise independent hash function $h_i : U \to [c]$, and each element in y satisfies:

$$y_{i,u} = \sum_{v|h_i(v)=u} x_v$$

Every update (v, t) modifies the sketch Y as follows (Figure 1)

$$\forall i \in [r], \quad y_{i,h_i(v)} \mathrel{+}= t.$$

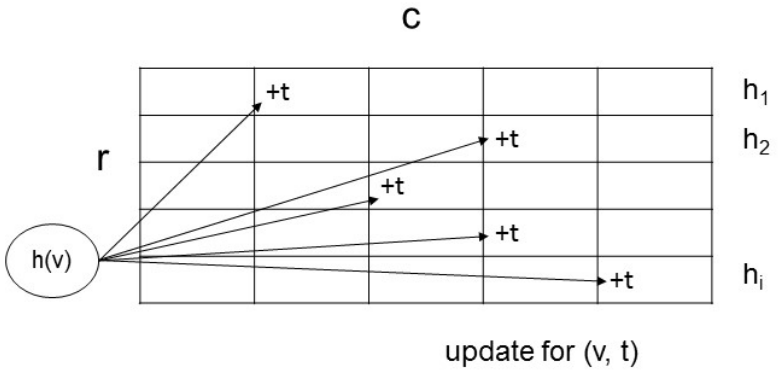

Figure 1: Updating count-min sketch

Given the sketch Y, we define $\tilde{x_v}^{(i)} = y_{i,h_i(v)}$ as the estimate of x_v from i^{th} row. It is straightforward to see that we always overestimate the value because $x_v \geq 0 \ \forall v$ under the strict turnstile model. Therefore, to get a better estimate for x_v, we compute $\tilde{x_v}$ as

$$\tilde{x_v} = \min_i \tilde{x_v}^{(i)}.$$

Claim 1. *(Informal)* $Z_i = \tilde{x_v}^{(i)} - x_v$ *is small* $\left(\leq 2\frac{||\mathbf{x}||_1}{c}\right)$ *with probability* $\geq \frac{1}{2}$.

Proof. In order to prove the above claim, we show an upper bound on $\mathbb{E}[Z]$, and then apply markov's inequality.

$$\mathbb{E}[\tilde{x_v}^{(i)} - x_v] = \mathbb{E}\left[\sum_{v' \neq v} x_{v'} \cdot I_{h_i(v')=h_i(v)}\right] \tag{1}$$

$$= \sum_{v' \neq v} x_{v'} \cdot \mathbb{E}[I_{h_i(v')=h_i(v)}] \tag{2}$$

$$= \sum_{v' \neq v} x_{v'} \cdot \frac{1}{c} \tag{3}$$

$$\leq \frac{\|\mathbf{x}\|_1}{c} \tag{4}$$

$I_{h_i(v')=h_i(v)}$ denotes the indicator function which is 1 *iff* $h_i(v')$ and $h_i(v)$ collide. Equation (2) to (3) follow from pairwise independence of h_i.

□

Now we use markov's inequality to obtain,

$$\mathbb{P}\left(\mathbb{E}[\tilde{x_v}^{(i)} - x_v] \leq \frac{2\|\mathbf{x}\|_1}{c}\right) \leq \frac{1}{2}$$
$$\Rightarrow \mathbb{P}\left(\mathbb{E}[\tilde{x_v} - x_v] \leq \frac{2\|\mathbf{x}\|_1}{c}\right) \leq 1 - \frac{1}{2^r} = 1 - \delta.$$

Let $c = \dfrac{2}{\epsilon\alpha}$, we get

$$\begin{aligned}\mathbb{P}\left(\mathbb{E}[\tilde{x_v} - x_v] \leq \frac{2\|\mathbf{x}\|_1}{c}\right) &= \mathbb{P}\left(\mathbb{E}[\tilde{x_v} - x_v] \leq \epsilon\alpha\|\mathbf{x}\|_1\right) \\ &= \mathbb{P}(\mathbb{E}[\tilde{x_v} - x_v] \leq \epsilon\alpha m) \\ &\leq 1 - \delta\end{aligned}$$

Now to find the (ϵ, δ)-heavy hitters (find $S \supseteq \text{HH}_\alpha$), we look at all $v \in U$, and take $S := \{v \mid \tilde{x_v} \geq \alpha m\}$. First, observe that if $x_v \geq \alpha m$, then $\tilde{x_v} \geq x_v \geq \alpha m$, therefore $v \in S$. Second, if $x_v < (1-\epsilon)\alpha m$, then v does not satisfy the criterion of belonging to (ϵ, δ)-HH_α, and we know from above inequalities that the probability $\tilde{x_v} \geq (1-\epsilon)\alpha m$ given $x_v < (1-\epsilon)\alpha m$ is at most $\delta = 1/2^r$. Therefore, the probability of any item being misclassified as $\epsilon - \text{HH}_\alpha$ is at most $1/2^r$, so using union bound, we could write that

$$\mathbb{E}[\text{No. of incorrect classifications}] \leq |U| \cdot \delta$$

Therefore, if we replace r from $\log \dfrac{1}{\delta}$ to $\log \dfrac{|U|}{\delta}$, we get $\delta' = \frac{\delta}{|U|}$, and $E[\text{No. of incorrect classifications}] \leq |U|\delta' = \delta$. Therefore, using markov inequality

$$\mathbb{P}(\mathbb{E}[\text{No. of incorrect classifications}] \geq 1) \leq \delta$$

So now we get the set S as $(\epsilon, \delta) - \text{HH}_\alpha$ heavy hitter approximation with probability $1-\delta$, and the time complexity is $O\left(\frac{1}{\epsilon\alpha}\log\frac{|U|}{\delta}\right)$.

3.1 Improvement of error bound

Previous analysis make no assumption on the data stream, however, we observe that many kinds of real data has skewed distribution, such as power distribution, where the count of i^{th} item $x_i = C \cdot i^{-s}$, where C is a constant, and $s > 1$. When we estimate the count of the item, and when it collides with the first k large items, our error estimation will be much better than previously described.

Claim 2. *(Informal) We can achieve error approximately* $C\frac{\alpha^{s-1}}{s-1}$.

Let $H \subseteq U$ denote the top $c/4$ items of $\mathbf{x}$. Let $\mathbf{x}_{-k}$ be the set of items except for the first k heavy items. The collision probability is:

$$\forall v \in U, \quad \mathbb{P}(h_i(v) \in \{h_i(v') \mid v' \in H, v' \neq v\}) \leq \frac{|H|}{c} = \frac{1}{4}. \tag{5}$$

Let error $Z = \sum_{v' \neq v, h_i(v') = h_i(v), v \notin H} x_v$. We have

$$\mathbb{E}[Z] = \frac{1}{c}\sum_{v' \neq v, v \notin H} x_v = \frac{\|\mathbf{x}_{-\frac{c}{4}}\|_1}{c}$$

$$\mathbb{P}\left(Z \geq 4\frac{\|\mathbf{x}_{-\frac{c}{4}}\|_1}{c}\right) \leq \frac{1}{4} \tag{6}$$

Also note

$$\|\mathbf{x}_{-k}\|_1 = C\sum_{k+1}^{\infty}\frac{1}{i^s} \leq C\int_k^\infty \frac{1}{i^s}di = C\frac{1}{s-1}\frac{1}{k^{s-1}}. \tag{7}$$

So the probability neither (5) or (6) happen is $\geq \frac{1}{2}$, with error $O\left(\frac{C}{c(s-1)}\left(\frac{4}{c}\right)^{s-1}\right)$, or $O\left(C\frac{\alpha^{s-1}}{s-1}\right)$ because $c = O\left(\frac{1}{\alpha}\right)$.

3.2 Improvement of time complexity

Now we have shown that with count-min sketch, we can compute ϵ heavy hitter $S \supseteq \{v \mid x_v \geq \alpha\|\mathbf{x}\|_1\}$, $|S| \leq \frac{2}{\alpha}$ with $O\left(\frac{1}{\alpha}\log|U|\right)$ space. Next, we try to improve the time complexity.

Observe that if v is heavy ($x_v \geq \alpha\|x\|_1$), for all $S \supseteq \{v\}$, we have $\sum_{v \in S} x_v \geq \alpha\|x\|_1$, or these sets are heavy as well.

We solve the heavy hitter problem using a divide and conquer procedure, using $\log |U|$ sketches. The first sketch will count the two subsets separated based on the difference of the first bit of each item. The second sketch count subsets on the difference of the first two bits within the four subsets. For sketch t, we have

$$y_{t,i,u} = \sum_{v:h_{t,i}} (\text{first } t \text{ bits of v} == u)$$

To get all the heavy hitters, we perform a binary search on these subsets from sketch 1 to $\log |U|$, as described before, at each level, if the subset S contains the heave hitter, the subset itself is heavy. We trace the heavy subset to the last sketch and output all the heavy single items. The time complexity in this case is $O\left(\frac{1}{\alpha} \log^2 |U|\right)$.

References

[MR] Rajeev Motwani, Prabhakar Raghavan Randomized Algorithms. *Cambridge University Press*, 0-521-47465-5, 1995.

[CM] G. Cormode and S. Muthukrishnan. Whats hot and whats not: Tracking most frequent items dynamically. *In Proceedings of ACM Principles of Database Systems*, pages 296-306, 2003.

[CM] G. Cormode and S. Muthukrishnan. An improved data stream summary: The count-min sketch and its applications. *Journal of Algorithms*, 55(1):5875, 2005.