CS 395T: Sublinear Algorithms Fall 2014

Lecture 4 — Sept. 9, 2014

Prof. Eric Price *Scribe: Zhao Song*

1 Overview

In today's lecture, we will discuss the following problems:

1. Count-Min [CM05]
2. Count-Sketch [CCF02]
3. Fourier analysis of Count-Sketch [MP14]

Motivation All these things are dominated by top few elements

1. URLS on the web
2. Words in Shakespeare
3. IPs in router

2 Preliminaries

Definition 2.1. *Let x denote a vector of length n, $x_{(i)}$ = the i^{th} largest term.*

Definition 2.2. *Let $x_{(1,\cdots,k)}$ denote the vector that zeros every coordinates but takes top-k largest terms, and $x_{(k+1,\cdots,n)} = x - x_{(1,\cdots,k)}$.*

Definition 2.3. *Let $H = (1,\cdots,k)$ (Head set) and $T = (k+1,\cdots,n)$ (Tail set).*

Observation Many kinds of data come from power law distributions, such as $x_{(i)} \propto i^{-\alpha}$, where $\alpha \approx .8$.

Claim 2.4. $\frac{\|x_{(k+1,\cdots,n)}\|_2}{\|x\|_2} \propto \frac{1}{k^{\frac{2\alpha-1}{2}}}$

Make a sketch $A \in \mathbb{R}^{m\times m}$, and $m \approx k\log^c n$.

Given $y = Ax$, we want an estimate $\hat{x}$ of x such that

$$\|\hat{x} - x\| \leq c\|x - x_{(1,\cdots,k)}\| = c\|x_{(k+1,\cdots,n)}\|$$

where c is a constant factor. (What norm do you care about? You can choose $\ell_1, \ell_2, \ell_\infty$...)

For ℓ_1 norm:

$$\|\hat{x} - x\|_\infty \leq \frac{\epsilon}{k}\|x_{(k+1,\cdots,n)}\|_1 \implies \|\overline{x} - x\|_1 \leq (1+\epsilon)\|x_{(k+1,\cdots,n)}\|_1, where\ m \approx O(\frac{k}{\epsilon}\log n)$$

where $\overline{x}$ contains the largest k coordinates of $\hat{x}$. Similarly, for ℓ_2, we have

$$\|\hat{x} - x\|_\infty \leq \frac{\epsilon}{\sqrt{k}}\|x_{(k+1,\cdots,n)}\|_2 \implies \|\overline{x} - x\|_2 \leq (1+\epsilon)\|x_{(k+1,\cdots,n)}\|_2, where\ m \approx O(\frac{k}{\epsilon^2}\log n)$$

where $\overline{x}$ contains the largest k coordinates of $\hat{x}$. (You will prove these in your homework.)

What's the difference between these norms? Well, for power law distributions we get:

Claim 2.5. $\frac{\|x_{(k+1,\cdots,n)}\|_1}{\|x\|_1} \approx \frac{1}{k^{\alpha-1}}$ *this is true if* $\alpha > 1$

Claim 2.6. $\frac{\|x_{(k+1,\cdots,n)}\|_1}{\|x\|_1} \approx \frac{1}{k^{\frac{2\alpha-1}{2}}}$ *this is true if* $\alpha > \frac{1}{2}$

Note that in the common case of $0.5 \leq \alpha \leq 1$, only the ℓ_2 guarantee is useful.

In this class we will give a $\Theta(n \log n)$ time algorithm to do recovery. Next class we will improve this to $o(n)$ time($\approx \Theta(k \cdot \text{poly}(\log n))$).

3 Count-Min

Some other lecture notes also provide the details of Count-Min.

Lecture 3 of Course "Sublinear Algorithms for Big Data" at the University of Buenos Aires,

http://grigory.github.io/files/teaching/sublinear-big-data-3.pdf .

Lecture 5 of Course "Algorithm for Big Data" at Harvard University,

http://people.seas.harvard.edu/ minilek/cs229r/lec/lec5.pdf .

Sketch:

We think about storing a "table" with R rows and B columns, and a counter $y_v^{(u)}$ for each cell (u, v) of the table.

1. Choose R pair-wise independent hash functions $h_1, h_2, \cdots, h_R : [n] \to [B]$.
2. For each hash function/row h_u, we need B counters.
3. $\forall u \in [R], \forall v \in [B], y_v^{(u)} = \sum\limits_{i, h_u(i)=v} x_i$.

This is a linear function of x, so it can be expressed as a matrix.

Recovery Algorithm

Given y, we recover our estimate $\hat{x}$ of x by:

1. In each row, estimate $\hat{x}_i^{(u)} = y^{(u)}_{h_u(i)}$.
2. Overall, estimate $\hat{x}_i = \min_u \hat{x}_i^{(u)}$.

The intution of this algorithm is trying to separate large terms from small terms.

Analysis:

Let $H = (1, \cdots, k)$ and $T = (k+1, \cdots, n)$. For a particular hash function h_u:

$$\begin{aligned}
\|\hat{x}_i^{(u)} - x_i\| &= \sum_{j \in H, h_u(i)=h_u(j)} x_j + \sum_{j \in T, h_u(i)=h_u(j)} x_j \\
&\leq \underbrace{0}_{with\ probability\ 1-\frac{k}{B}} + \underbrace{\|x_T\|_1/B}_{in\ expectation} \\
&\leq \underbrace{0}_{with\ probability\ \frac{9}{10}} + \underbrace{\frac{\|x_T\|_1}{k}}_{with\ probability\ \frac{9}{10}}
\end{aligned}$$

where we set $B = 10k$.

Thus for each u and i, by a union bound we have

$$\|\hat{x}_i^{(u)} - x_i\| \leq \frac{\|x_T\|_1}{k}\ with\ probability\ \frac{8}{10}.$$

then it implies that

$$\hat{x}_i = \min_u \hat{x}_i^{(u)} \leq x_i + \frac{\|x_T\|_1}{k}\ with\ probability\ 1 - (\frac{1}{5})^R$$

Choose $R = O(\log n)$, then

$$BR = O(k \log n),\ 1 - (\frac{1}{5})^R = 1 - n^{-c},\ where\ c\ is\ a\ constant\ value.$$

What if some coordinates are negative?

For some error $\sigma = O(\|x_T\|_1/k)$, we have that $Pr[|\hat{x}_i^{(u)} - x_i| \leq \sigma] \geq \frac{4}{5}$. Then after R samples, with $1 - e^{-O(R)}$ probability we will have that at least $\frac{n}{2}$ of the $\hat{x}_i^{(u)}$ will land in $x_i \pm \sigma$. Their median then to land in that region.

4 Count-Sketch

4.1 Other references

Some other lecture notes also provide the details of Count-Sketch.

Lecture 3 of Course “Sublinear Algorithms for Big Data” at the University of Buenos Aires,

http://grigory.github.io/files/teaching/sublinear-big-data-3.pdf .

Lecture 5 of Course “Algorithm for Big Data” at Harvard University,

http://people.seas.harvard.edu/ minilek/cs229r/lec/lec5.pdf .

4.2 Setup

One issue with count-min is that if the vector is positive everywhere, then all the errors go in the same direction. The idea of count-sketch is to introduce random signs in the summation, so that the errors tend to cancel each other out. This converts the bound from ℓ_1 to ℓ_2, which is more useful.

Sketch:

We think about storing a “table” with R rows and B columns, and a counter $y_v^{(u)}$ for each cell (u, v) of the table.

1. Choose R pair-wise independent hash functions $h_1, h_2, \cdots, h_R : [n] \to [B]$ and $s_1, \ldots, s_R : [n] \to \{\pm 1\}$.
2. For each hash function/row h_u, we need B counters.
3. $\forall u \in [R], \forall v \in [B], y_v^{(u)} = \sum\limits_{i, h_u(i)=v} s_u(i) x_i$.

This is a linear function of x, so it can be expressed as a matrix.

Recovery Algorithm

Given y, we recover our estimate $\hat{x}$ of x by:

1. In each row, estimate $\hat{x}_i^{(u)} = s_u(i) y_{h_u(i)}^{(u)}$.
2. Overall, estimate $\hat{x}_i = \text{median}_u \hat{x}_i^{(u)}$.

The only difference from Count-Min is the introduction of the signs s_u, and the use of the median for estimation.

Analysis:

Let’s bound the term $\|\hat{x}_i^{(u)} - x_i\|$ for every $u \in [R]$:

$$\begin{aligned} \|\hat{x}_i^{(u)} - x_i\| &= \sum_{j \in H, h_u(i)=h_u(j)} x_j + \sum_{j \in T, h_u(i)=h_u(j)} x_j \\ &\leq \underbrace{0}_{with\ probability\ 1-\frac{k}{B}} + \underbrace{\Delta}_{\underset{h}{E}\underset{s}{E}[\Delta^2]=E\sum x_j^2=\|x_T\|_2^2/B} \end{aligned}$$

Then, we have

$$\begin{aligned}
&\Delta^2 \leq \frac{\|x_T\|_2^2}{k} \textit{ with probability } \frac{9}{10}\\
\implies &\|\hat{x}_i^{(u)} - x_i\| \leq \frac{\|x_T\|_2}{\sqrt{k}} \textit{ with probability } \frac{4}{5}\\
\implies &\|\hat{x}_i - x_i\| \leq \frac{\|x_T\|_2}{\sqrt{k}} \textit{ with probability } 1 - n^{-c} \textit{ (by setting } R = O(\log n))
\end{aligned}$$

5 Fourier Analysis of [MP14]

You can actually give a tighter analysis of Count-Sketch, which shows that *most* coordinates are estimated to higher precision, if your hash functions are fully independent. As we described in an earlier class, the assumption of fully independent hash functions is unfortunate, but it can be justified using cryptographic hash functions and computational assumptions on the adversarial input, or assuming the input has high entropy.

Note that the details of this analysis also can be found in Eric Price's presentation slide of SODA'2015. Here is the link : http://www.cs.utexas.edu/ ecprice/slides/concentration-slides.pdf.

Theorem 5.1. *Assume that h and s are fully independent hash functions, and consider the output $\hat{x}$ of Count-Sketch. Then $\forall t \leq R$, we have*

$$|\hat{x}_i - x_i| \leq \sqrt{\frac{t}{R}} \cdot \frac{\|x_T\|_2}{\sqrt{k}}$$

with probability $1 - e^{-\Omega(t)}$.

This implies that $\mathbb{E}[\hat{x}_i - x_i] \leq \frac{1}{\sqrt{R}} \cdot \frac{\|x_T\|_2}{\sqrt{k}}$ after excluding $e^{-\Omega(R)}$ events.

Before we get into the prove, let's look at a simpler problem:

5.1 Estimating a symmetric random variable's point of symmetry

Suppose we have an unknown distribution $\mathcal{X}$ over R, which is symmetric about unknown μ. How can we best recover μ from a set of samples $x_1, \ldots, x_R \sim \mathcal{X}$?

For example, you might consider the following distribution:

$$\mathcal{X} = \begin{cases} \textit{mean } \mu, \textit{ standard deviation } \sigma & \text{with probability } \frac{1}{2} \\ \mu \pm \infty & \text{with probability } \frac{1}{2} \end{cases}$$

1. Mean
   - (a) Converges to μ as $\frac{\sigma}{\sqrt{R}}$
   - (b) No robustness to outliers
2. Median

(a) Extremely robust

(b) Doesn't necessarily converge to μ

We show that it will work if you take the median of pairwise means:

$$\underset{i\in\{1,3,5,\cdots\}}{\text{median}} \frac{x_i + x_{i+1}}{2}$$

which converges as $O(\sigma/\sqrt{R})$. (Similar to Hodges-Lehmann estimator[1])

Why does median converge for $(\mathcal{X}+\mathcal{X})/2$?

1. Without loss of generality, assume that $\mu = 0$.
2. Define the Fourier transform $\mathcal{F}_\mathcal{X}$ of $\mathcal{X}$: $\mathcal{F}_\mathcal{X}(t) = \mathbb{E}_{x\sim\mathcal{X}}[\cos(\tau x t)]$, where $\tau = 2\pi \approx 6.28$.
3. Convolution $\iff$ multiplication: $\mathcal{F}_{\mathcal{X}+\mathcal{X}}(t) = (\mathcal{F}_\mathcal{X}(t))^2 \geq 0$ for all t.

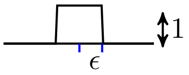
Theorem 5.2. *Let $\mathcal{Y}$ be symmetric about 0 with $\mathcal{F}_\mathcal{Y}(t) \geq 0$ $\forall t$ and $\mathbb{E}[Y^2] = \sigma^2$. Then $\forall\epsilon \leq 1$*

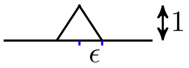
$$Pr[|y| \leq \epsilon\sigma] \gtrsim \epsilon.$$

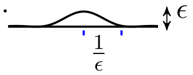
Let's consider the proof when $\sigma = 1$.

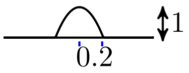
Proof.

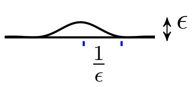
$$\mathcal{F}_\mathcal{Y}(t) = \mathbb{E}[\cos(\tau y t)] \geq 1 - \frac{\tau^2}{2}t^2$$

$Pr[|y| \leq \epsilon] = \mathcal{Y}\cdot$ 

$\geq \mathcal{Y}\cdot$ 

$= \mathcal{F}_\mathcal{Y}\cdot$ 

$\geq$ 

$\cdot$ 

$\gtrsim \epsilon$

where we use that the Fourier transform of the triangle function is $(\frac{\sin x}{x})^2$ is positive.

□

[1] http://en.wikipedia.org/wiki/HodgesLehmann_estimator

References

[CCF02] Moses Charikar, Kevin Chen, and Martin Farach-Colton. Finding Frequent Items in Data Streams. *ICALP*, 2002.

[CM05] Graham Cormode and S. Muthukrishnan. Summarizing and Mining Skewed Data Streams. *SDM*, 2005.

[MP14] Gregory T. Minton and Eric Price. Improved Concentration Bounds for Count-Sketch *SODA(best student paper)* 2014.