CS 598CSC: Algorithms for Big Data	**Lecture date: Sept 11, 2014**
Instructor: Chandra Chekuri	**Scribe: Chandra Chekuri**

The Misra-Greis deterministic counting guarantees that all items with frequency $> F_1/k$ can be found using $O(k)$ counters and an update time of $O(\log k)$. Setting $k = 1/\epsilon$ one can view the algorithm as providing an additive ϵF_1 approximation for each f_i. However, the algorithm does not provide a sketch. One advantage of linear sketching algorithms is the ability to handle deletions. We now discuss two sketching algorithms that have a found a number of applications. These sketches can be used to for estimating point queries: after seeing a stream σ over items in $[n]$ we would like to estimate f_i the frequency of $i \in [n]$. More generally, in the turnstile model, we would like to estimate x_i for a given $i \in [n]$. We can only guarantee the estimate with an *additive* error.

1 CountMin Sketch

We firt describe the simpler CountMin sketch. The sketch maintains several counters. The counters are best visualized as a rectangular array of width w and depth d. With each row i we have a hash function $h_i : [n] \to [w]$ that maps elements to one of w buckets.

```
CountMin-Sketch(w, d):

h_1, h_2, ..., h_d are pair-wise independent hash functions from [n] → [w].
While (stream is not empty) do
   a_t = (i_t, Δ_t) is current item
   for ℓ = 1 to d do
         C[ℓ, h_ℓ(i_j)] ← C[ℓ, h_ℓ(i_j)] + Δ_t
endWhile
For i ∈ [n] set x̃_i = min_{ℓ=1}^{d} C[ℓ, h_ℓ(i)].
```

The counter $C[\ell, j]$ simply counts the sum of all x_i such that $h_\ell(i) = j$. That is,

$$C[\ell, j] = \sum_{i : h_\ell(i) = j} x_i.$$

Exercise: CountMin is a linear sketch. What are the entries of the projection matrix?

We will analyze the sketch in the strick turnstile model where $x_i \geq 0$ for all $i \in [n]$; note that Δ_t we be negative.

Lemma 1 *Let $d = \Omega(\log \frac{1}{\delta})$ and $w > \frac{2}{\epsilon}$. Then for any fixed $i \in [n]$, $x_i \leq \tilde{x}_i$ and*

$$\Pr[\tilde{x}_i \geq x_i + \epsilon \|\mathbf{x}\|_1] \leq \delta.$$

Proof: Fix $i \in [n]$. Let $Z_\ell = C[\ell, h_\ell(i)]$ be the value of the counter in row ℓ to which i is hashed to. We have

$$\mathbf{E}[Z_\ell] = x_i + \sum_{i' \neq i} \Pr[h_\ell(i') = h_\ell(i)] x_{i'} = x_i + \sum_{i' \neq i} \frac{1}{w} x_{i'} \leq x_i + \frac{\epsilon}{2} \|\mathbf{x}\|_1.$$

Note that we used pair-wise independence of h_ℓ to conclude that $\Pr[h_\ell(i') = h_\ell(i)] = 1/w$.

By Markov's inequality (here we are using non-negativity of $\mathbf{x}$),

$$\Pr[Z_\ell > x_i + \epsilon\|\mathbf{x}\|_1] \leq 1/2.$$

Thus

$$\Pr[\min_\ell Z_\ell > x_i + \epsilon\|\mathbf{x}\|_1] \leq 1/2^d \leq \delta.$$

□

Remark: By choosing $\delta = \Omega(\log n)$ we can ensure with probability at least $(1 - 1/\text{poly}(n))$ that $\tilde{x}_i - x_i \leq \epsilon\|\mathbf{x}\|_1$ for *all* $i \in [n]$.

Exercise: For general turnstile streams where $\mathbf{x}$ can have negative entries we can take the median of the counters. For this estimate you should be able to prove the following.

$$\Pr[|\tilde{x}_i - x_i| \geq 3\epsilon\|\mathbf{x}\|_1] \leq \delta^{1/4}.$$

2 Count Sketch

Now we discuss the closely related Count sketch which also maintains an array of counters parameterized by the width w and depth d.

COUNT-SKETCH(w, d):

$h_1, h_2, \ldots, h_d$ are pair-wise independent hash functions from $[n] \to [w]$.
$g_1, g_2, \ldots, g_d$ are pair-wise independent hash functions from $[n] \to \{-1, 1\}$.
While (stream is not empty) do
  $a_t = (i_t, \Delta_t)$ is current item
  for $\ell = 1$ to d do
    $C[\ell, h_\ell(i_j)] \leftarrow C[\ell, h_\ell(i_j)] + g(i_t)\Delta_t$
endWhile
For $i \in [n]$ set $\tilde{x}_i = \text{median}\{g_1(i)C[1, h_1(i)], g_2(i)C[2, h_2(i), \ldots, g_d(i)C[d, h_d(i)]\}$.

Exercise: CountMin is a linear sketch. What are the entries of the projection matrix?

Lemma 2 *Let $d \geq \log\frac{1}{\delta}$ and $w > \frac{3}{\epsilon^2}$. Then for any fixed $i \in [n]$, $\mathbf{E}[\tilde{x}_i] = x_i$ and*

$$\Pr[|\tilde{x}_i - x_i| \geq \epsilon\|\mathbf{x}\|_2] \leq \delta.$$

Proof: Fix an $i \in [n]$. Let $Z_\ell = g_\ell(i)C[\ell, h_\ell(i)]$. For $i' \in [n]$ let $Y_{i'}$ be the indicator random variable that is 1 if $h_\ell(i) = h_\ell(i')$; that is i and i' collide in h_ℓ. Note that $\mathbf{E}[Y_{i'}] = \mathbf{E}[Y_{i'}^2] = 1/w$ from the pairwise independence of h_ℓ. We have

$$Z_\ell = g_\ell(i)C[\ell, h_\ell(i)] = g_\ell(i)\sum_{i'} g_\ell(i')x_{i'}Y_{i'}$$

Therefore,

$$\mathbf{E}[Z_\ell] = x_i + \sum_{i' \neq i} \mathbf{E}[g_\ell(i)g_\ell(i')Y_{i'}]x_{i'} = x_i,$$

because $\mathbf{E}[g_\ell(i)g_\ell(i')] = 0$ for $i \neq i'$ from pairwise independence of g_ℓ and $Y_{i'}$ is independent of $g_\ell(i)$ and $g_\ell(i')$. Now we upper bound the variance of Z_ℓ.

$$\begin{aligned}
\mathbf{Var}[Z_\ell] &= \mathbf{E}\left[(\sum_{i' \neq i} g_\ell(i)g_\ell(i')Y_{i'}x_{i'})^2\right] \\
&= \mathbf{E}\left[\sum_{i' \neq i} x_{i'}^2 Y_{i'}^2 + \sum_{i' \neq i''} x_{i'}x_{i''}g_\ell(i')g_\ell(i'')Y_{i'}Y_{i''}\right] \\
&= \sum_{i' \neq i} x_{i'}^2 \mathbf{E}[Y_{i'}^2] \\
&\leq \|\mathbf{x}\|_2^2/w.
\end{aligned}$$

Using Chebyshev,

$$\Pr[|Z_\ell - x_i| \geq \epsilon\|\mathbf{x}\|_2] \leq \frac{\mathbf{Var}[Z_\ell]}{\epsilon^2\|\mathbf{x}\|_2^2} \leq \frac{1}{\epsilon^2 w} \leq 1/3.$$

Now, via the Chernoff bound,

$$\Pr\left[|\text{median}\{Z_1, \ldots, Z_d\} - x_i| \geq \epsilon\|\mathbf{x}\|_2\right] \leq e^{-cd} \leq \delta.$$

Thus choosind $d = O(\log n)$ and taking the median guarantees the desired bound with high probability. □

Remark: By choosing $\delta = \Omega(\log n)$ we can ensure with probability at least $(1 - 1/\text{poly}(n))$ that $|\tilde{x}_i - x_i| \leq \epsilon\|\mathbf{x}\|_2$ for *all* $i \in [n]$.

3 Applications

Count and CountMin sketches have found a number of applications. Note that they have a similar structure though the guarantees are different. Consider the problem of estimating frequency moments. Count sketch outputs an estimate $\tilde{f}_i$ for f_i with an additive error of $\epsilon\|\mathbf{f}\|_2$ while CountMin guarantees an additive error of $\epsilon\|\mathbf{f}\|_1$ which is always larger. CountMin provides a one-sided error when $\mathbf{x} \geq 0$ which has some benefits. CountMin uses $O(\frac{1}{\epsilon}\log\frac{1}{\delta})$ counters while Count sketch uses $O(\frac{1}{\epsilon^2}\log\frac{1}{\delta})$ counters. Note that the Misra-Greis algorithm uses $O(1/\epsilon)$-counters.

3.1 Heavy Hitters

We will call an index i an α-HH (for heavy hitter) if $x_i \geq \alpha\|x\|_1$ where $\alpha \in (0, 1]$. We would like to find S_α, the set of all α-heavy hitters. We will relax this assumption to output S such that

$$S_\alpha \subseteq S \subseteq S_{\alpha-\epsilon}.$$

Here we will assume that $\alpha < \alpha$ for otherwise the approximation does not make sense.

Suppose we used CountMin sketch with $w = 2/\epsilon$ and $\delta = c/n$ for sufficiently large c. Then, as we saw, with probability at least $(1 - 1/\text{poly}(n))$, for all $i \in [n]$,

$$x_i \leq \tilde{x}_i \leq x_i + \epsilon\|\mathbf{x}\|_1.$$

Once the sketch is computed we can simply go over all i and add i to S if $\tilde{x}_i \geq \alpha\|\mathbf{x}\|_1$. It is easy to see that S is the desired set.

Unfortunately the computation of S is expensive. The sketch has $O(\frac{1}{\epsilon}\log n)$ counters and processing each i takes time proportional to the number of counters and hence the total time is $O(\frac{1}{\epsilon}n\log n)$ to output a set S of size $O(\frac{1}{\alpha})$. It turns that by keeping additional information in the sketch in a hierarchical fashion one can cut down the time to be proportional to $O(\frac{1}{\alpha}\text{polylog}(n)))$.

3.2 Range Queries

In several application the range $[n]$ corresponds to an actual total ordering of the items. For instance $[n]$ could represent the discretization of time and $\mathbf{x}$ corresponds to the signal. In databases $[n]$ could represent ordered numerical attributes such as age of a person, height, or salary. In such settings range queries are very useful. A range query is an interval of the form $[i,j]$ where $i,j \in [n]$ and $i \leq j$. The goal is to output $\sum_{i\leq \ell \leq j} x_i$. Note that there are $O(n^2)$ potential queries.

There is a simple trick to solve this using the sketches we have seen. An interval $[i,j]$ is a *dyadic* interval/range if $j-i+1$ is 2^k and 2^k divides $i-1$. Assume n is a power of 2. Then the dyadic intervals of length 1 are $[1,1],[2,2],\ldots,[n,n]$. Those of length 2 are $[1,2],[3,4],\ldots$ and of length 4 are $[1,4],[5,8],\ldots$.

Claim 3 *Every range $[i,j]$ can be expressed as a disjoint union of at most $2\log n$ dyadic ranges.*

Thus it suffices to maintain accurate point queries for the dyadic ranges. Note that there are at most $2n$ dyadic ranges. They fall into $O(\log n)$ groups based on length; the ranges for a given length partition the entire interval. We can keep a separate CountMin sketch for the $n/2^i$ dyadic intervals of length i ($i = 0$ corresponds to the sketch for point queries). Using these $O(\log n)$ CountMin sketches we can answer any range query with an additive error of $\epsilon\|\mathbf{x}\|_1$. Note that a range $[i,j]$ is expressed as the sum of $2\log n$ point queries each of which has an additive error. So ϵ' for the sketches has to be chosen to be $\epsilon/(2\log n)$ to ensure an additive error of $\epsilon\|\mathbf{x}\|_1$ for the range queries.

By choosing $d = O(\log n)$ the error probability for all point queries in all sketches will be at most $1/\text{poly}(n)$. This will guarantee that all range queries will be answered to within an additive $\epsilon\|\mathbf{x}\|_1$. The total space will be $O(\frac{1}{\epsilon}\log^3 n)$

3.3 Sparse Recovery

Let $\mathbf{x} \in \mathbb{R}^n$ be a vector. Can we approximate $\mathbf{x}$ by a sparse vector $\mathbf{z}$? By sparse we mean that $\mathbf{z}$ has at most k non-zero entries for some given k (this is the same as saying $\|\mathbf{z}\|_0 \leq k$). A reasonable way to model this is to ask for computing the error

$$\text{err}_p^k(\mathbf{x}) = \min_{\mathbf{z}:\|\mathbf{z}\|_0\leq k} \|\mathbf{x}-\mathbf{z}\|_p$$

for some p. A typical choice is $p=2$. It is easy to see that the optimum $\mathbf{z}$ is obtained by restricting $\mathbf{x}$ to its k largest coordinates (in absolute value). The question we ask here is whether we can estimate $\text{err}_2^k(\mathbf{x})$ efficiently in a streaming fashion. For this we use the Count sketch. Recall that by choosing $w = 3/\epsilon^2$ and $d = \Theta(\log n)$ the sketch ensures that with high probability,

$$\forall i \in [n], \quad |\tilde{x}_i - x_i| \leq \epsilon\|\mathbf{x}\|_2.$$

One can in fact show a generalization.

Lemma 4 *Count-Sketch with $w = 3k/\epsilon$ and $d = O(\log n)$ ensures that*

$$\forall i \in [n], \quad |\tilde{x}_i - x_i| \le \frac{\epsilon}{\sqrt{k}} err_2^k(\mathbf{x}).$$

Proof: Let $S = \{i_1, i_2, \ldots, i_k\}$ be the indices of the largest coordinates in $\mathbf{x}$ and let $\mathbf{x}'$ be obtained from $\mathbf{x}$ by setting entries of $\mathbf{x}$ to zero for indices in S. Note that $\mathrm{err}_2^k(\mathbf{x}) = \|\mathbf{x}'\|_2$. Fix a coordinate i. Consider row ℓ and let $Z_\ell = g_\ell(i)C[\ell, h_\ell(i)]$ as before. Let A_ℓ be the event that there exists an index $t \in S$ such that $h_\ell(i) = h_\ell(t)$; that is any "big" coordinate collides with i under h_ℓ. Note that $\Pr[A_\ell] \le \sum_{t \in S} \Pr[h_\ell(i) = \Pr[h_\ell(t)] \le |S|/w \le \epsilon/3$ by pair-wise independence of h. Now we estimate

$$\begin{aligned}
\Pr[|Z_\ell - x_i| \ge \frac{\epsilon}{\sqrt{k}} \mathrm{err}_2^k(\mathbf{x})] &= \Pr[|Z_\ell - x_i| \ge \frac{\epsilon}{\sqrt{k}} \|\mathbf{x}'\|_2] \\
&= \Pr[A_\ell] \cdot \Pr[|Z_\ell - x_i| \ge \frac{\epsilon}{\sqrt{k}} \|\mathbf{x}'\|_2] + \Pr[|Z_\ell - x_i| \ge \frac{\epsilon}{\sqrt{k}} \|\mathbf{x}'\|_2 \mid \neg A_\ell] \\
&\le \Pr[A_\ell] + 1/3 < 1/2.
\end{aligned}$$

□

Now let $\tilde{\mathbf{x}}$ be the approximation to $\mathbf{x}$ that is obtained from the sketch. We can take the k largest coordinates of $\tilde{\mathbf{x}}$ to form the vector $\mathbf{z}$ and output $\mathbf{z}$. We claim that this gives a good approximation to $\mathrm{err}_2^k(\mathbf{x})$. To see this we prove the following lemma.

Lemma 5 *Let $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ such that*

$$\|\mathbf{x} - \mathbf{y}\|_\infty \le \frac{\epsilon}{\sqrt{k}} err_2^k(\mathbf{x}).$$

Then,

$$\|\mathbf{x} - \mathbf{z}\|_2 \le (1 + 5\epsilon) err_2^k(\mathbf{x}),$$

where $\mathbf{z}$ is the vector obtained as follows: $\mathbf{z}_i = \mathbf{y}_i$ for $i \in T$ where T is the set of k largest (in absolute value) indices of $\mathbf{y}$ and $\mathbf{z}_i = 0$ for $i \notin T$.

Proof: Let $t = \frac{1}{\sqrt{k}} \mathrm{err}_2^k(\mathbf{x})$ to help ease the notation. Let S be the index set of the largest coordinates of $\mathbf{x}$. We have,

$$(\mathrm{err}_2^k(\mathbf{x}))^2 = kt^2 = \sum_{i \in [n] \setminus S} x_i^2 = \sum_{i \in T \setminus S} x_i^2 + \sum_{i \in [n] \setminus (S \cup T)} x_i^2.$$

We write:

$$\begin{aligned}
\|\mathbf{x} - \mathbf{z}\|_2^2 &= \sum_{i \in T} |x_i - z_i|^2 + \sum_{i \in S \setminus T} |x_i - z_i|^2 + \sum_{i \in [n] \setminus (S \cup T)} x_i^2 \\
&= \sum_{i \in T} |x_i - y_i|^2 + \sum_{i \in S \setminus T} x_i^2 + \sum_{i \in [n] \setminus (S \cup T)} x_i^2.
\end{aligned}$$

We treat each term separately. The first one is easy to bound.

$$\sum_{i \in T} |x_i - y_i|^2 \le \sum_{i \in T} \epsilon^2 t^2 \le \epsilon^2 k t^2.$$

The third term is common to $\|\mathbf{x}-\mathbf{z}\|_2$ and $\mathrm{err}_2^k(\mathbf{x})$. The second term is the one to care about.

Note that S is set of k largest coordinates in $\mathbf{x}$ and T is set of k largest coordinates in $\mathbf{y}$. Thus $|S \setminus T| = |T \setminus S|$, say their cardinality is $\ell \geq 1$. Since $\mathbf{x}$ and $\mathbf{y}$ are close in ℓ_∞ norm (that is they are close in each coordinate) it must mean that the coordinates in $S \setminus T$ and $T \setminus S$ are roughly the same value in $\mathbf{x}$. More precisely let $a = \max_{i \in S \setminus T} |x_i|$ and $b = \min_{i \in T \setminus S} |x_i|$. We leave it as an exercise to the reader to argue that that $a \leq b + 2\epsilon t$ since $\|\mathbf{x}-\mathbf{y}\|_\infty \leq \epsilon t$.

Thus,

$$\sum_{i \in S \setminus T} x_i^2 \leq \ell a^2 \leq \ell(b+2\epsilon t)^2 \leq \ell b^2 + 4\epsilon k t b + 4k\epsilon^2 t^2.$$

But we have

$$\sum_{i \in T \setminus S} x_i^2 \geq \ell b^2.$$

Putting things together,

$$\begin{aligned}
\|\mathbf{x}-\mathbf{z}\|_2^2 &\leq \ell b^2 + 4\epsilon k t b + \sum_{i \in [n] \setminus (S \cup T)} x_i^2 + 5k\epsilon^2 t^2 \\
&\leq \sum_{i \in T \setminus S} x_i^2 + \sum_{i \in [n] \setminus (S \cup T)} x_i^2 + 4\epsilon(\mathrm{err}_2^k(\mathbf{x}))^2 + 5\epsilon^2(\mathrm{err}_2^k(\mathbf{x}))^2 \\
&\leq (\mathrm{err}_2^k(\mathbf{x}))^2 + 9\epsilon(\mathrm{err}_2^k(\mathbf{x}))^2.
\end{aligned}$$

The lemma follows by by the fact that for sufficiently small ϵ, $\sqrt{1+9\epsilon} \leq 1+5\epsilon$.

□

Bibliographic Notes: Count sketch is by Charikar, Chen and Farach-Colton [1]. CountMin sketch is due to Cormode and Muthukrishnan [4]; see the papers for several applications. Cormode's survey on sketching in [2] has a nice perspective. See [3] for a comparative analysis (theoretical and experimenta) of algorithms for frinding frequent items. A deterministic variant of CountMin called CR-Precis is interesting; see `http://polylogblog.wordpress.com/2009/09/22/bite-sized-streams-cr-precis/` for a blog post with pointers and some comments. The applications are taken from the first chapter in the draft book by McGregor and Muthukrishnan.

References

[1] Moses Charikar, Kevin Chen, and Martin Farach-Colton. Finding frequent items in data streams. *Theoretical Computer Science*, 312(1):3–15, 2004.

[2] Graham Cormode, Minos N. Garofalakis, Peter J. Haas, and Chris Jermaine. Synopses for massive data: Samples, histograms, wavelets, sketches. *Foundations and Trends in Databases*, 4(1-3):1–294, 2012.

[3] Graham Cormode and Marios Hadjieleftheriou. Methods for finding frequent items in data streams. *VLDB J.*, 19(1):3–20, 2010.

[4] Graham Cormode and S. Muthukrishnan. An improved data stream summary: the count-min sketch and its applications. *J. Algorithms*, 55(1):58–75, 2005.