Sketching and streaming — Notes 3

Jelani Nelson
minilek@seas.harvard.edu

July 21, 2016

Today the focus of the lecture will be on **linear sketching**. Up until this point, we have focused on streaming algorithms in the so-called *insertion-only model*. Specifically, consider the scenario of a vector $x \in \mathbb{R}^n$ with n large, and x starting as the 0 vector. Then we see a sequence of updates (i, Δ) each causing the change $x_i \leftarrow x_i + \Delta$. You might imagine, for instance, that x has a coordinate for each string your search engine could possibly see as a query, and x_i is the number of times you've seen string i. In such an example, seeing string i corresponds to the update $(i, 1)$.

Then we have three popularly studied models:

1. **insertion-only:** Each update has $\Delta = 1$.
2. **strict turnstile:** Some updates Δ can be negative, but we are given the promise that $\forall i$, $x_i \geq 0$ at all times.
3. **general turnstile:** Anything goes. Updates can be negative, and entries in x can be negative as well.

In this lecture we will focus on algorithms for the strict and general turnstile models. All known algorithms for these models are actually linear sketches (which we shall see an explanation of shortly). It is in fact known [LNW14, AHLW16] that *any* algorithm in these two models can be converted into a linear sketch with only a logarithmic factor loss in space complexity.

So what is a *linear sketch*? It is an algorithm that maintains in memory Πx as x is updated, for some $\Pi \in \mathbb{R}^{m \times n}$ ($m \ll n$). Π may be deterministic, or it may be chosen at random from some probability distribution. Note we can

update $y = \Pi x$ easily after an update, since upon update (i, Δ), we simply need to add Δ times the ith column of Π to y since $\Pi(x+\Delta\cdot e_i) = \Pi x+\Delta\cdot\Pi_i$, where Π_i is the ith column of Π. Also note that if $\Pi \in \mathbb{R}^{m\times n}$, then naively it requires a lot of space to store (mn numbers). Thus, we will typically use linear sketches in which the entries of Π are specified implicitly, i.e. there is a small space algorithm that can retrieve $\Pi_{i,j}$ given any i, j.

1 Moment Estimation for $p = 2$

This problem was first investigated by Alon, Matias, and Szegedy [AMS99]. We let F_p denote $\|x\|_p^p = \sum_{i=1}^p |x_i|^p$. As usual, we would like a $(1 + \varepsilon)$-approximation to F_p with probability $2/3$. It turns out there is a transition point in the space complexity of F_p estimation.

- $0 \leq p \leq 2$: it is known that $poly(\varepsilon^{-1} \log n)$ words of space is achievable [AMS99, Ind06].
- For $p > 2$ and constant ε, it is known the space complexity is $n^{1-2/p}$ up to logarithmic factors [BJKS04, IW05]. That is, there are both upper and lower bounds.

We now look at the case $p = 2$, which is solved by the AMS sketch [AMS99]. Let $\sigma_1, \ldots, \sigma_m : [n] \to \{-1, 1\}$ be random hash functions each drawn independnetly from a 4-wise independent family. We need $O(\log n)$ bits, i.e. one machine word, to represent a single such hash function σ_i. Then set $y_i = \sum_{j=1}^n \sigma_i(j)x_j$. This means that our linear sketching matrix Π has ith row which is just the evaluation table of σ_i. Observe that it is easy to obtain any entry of Π in constant time by simply evaluating a hash function. Our algorithm is going to output $\frac{1}{m}\|y\|_2^2$ as our estimator of $\|x\|_2^2$.

Analysis. It holds that

$$\mathbb{E}y_i^2 = \mathbb{E}(\sum_{j=1}^n \sigma_{ij}x_j)^2 = \|x\|_2^2 + \mathbb{E}\sum_{j\neq j'} \sigma_{ij}\sigma_{ij'}x_jx_{j'} = \|x\|_2^2,$$

and thus $(1/m)\|y\|_2^2 = (1/m)\sum_{i=1}^m y_i^2$ is an unbiased estimator of $\|x\|_2^2$. Next we need to estimate the variance in order to apply Chebyshev's inequality.

Observe that

$$\begin{aligned}\mathbb{E}y_i^4 &= \mathbb{E}(\sum_j \sigma_{ij}x_j)^4 \\ &= \mathbb{E}\sum_{j_1,j_2,j_3,j_4\in[n]^4} \sigma_{ij_1}\sigma_{ij_2}\sigma_{ij_3}\sigma_{ij_4}x_{ij_1}x_{ij_2}x_{ij_3}x_{ij_4} \\ &= \sum_{j=1}^n x_j^4 + (1/2)\cdot\binom{4}{2}\sum_{j\neq j'} x_j^2x_{j'}^2,\end{aligned}$$

because when we take the expectation if a term in a summand appears with an odd exponent then its expectation is zero, so we are left only with the summands in which all terms appear an even number of times.

Also,

$$(\mathbb{E}\, y_i^2)^2 = \|x\|_2^4 = \sum_j x_j^4 + \sum_{j\neq j'} x_j^2x_{j'}^2,$$

and thus

$$Var[y_i^2] \leq 2\|x\|_2^4,$$

implying the variance of our estimator is at most $(2/m)\|x\|_2^4$. Thus by Chebyshev's inequality, the probability our estimator is outside of $[(1-\varepsilon)\|x\|_2^2, (1+\varepsilon)\|x\|_2^2]$, i.e. deviates from its expectation by more than $\varepsilon\|x\|_2^2$, is at most

$$\frac{2\|x\|_2^4}{m}\cdot\frac{1}{\varepsilon^2\|x\|_2^4} \leq 6/\varepsilon^2$$

for $m = \lceil 6/\varepsilon^2\rceil$.

2 Point query and heavy hitters

Here we discuss two different families of problems, called heavy hitters and point query. For both of these problems, we will consider x being updated in the turnstile model.

In the ℓ_1 *point query* problem, we must answer queries of the form query(i), for $i \in [n]$, with a value in the range $x_i \pm \varepsilon\cdot\|x\|_1$.

In ℓ_1 *heavy hitters*, there is only one query, and we must answer it with a set $L \subset [n]$ such that

1. $|x_i| \geq \varepsilon\|x\|_1 \Rightarrow i \in L$

2. $|x_i| < \frac{\varepsilon}{2}\|x\|_1 \Rightarrow i \notin L$

As an observation: if we can solve point query in small space then we can solve heavy hitters in small space as well (though not necessarily efficient run-time). To do this, we just run point query with $\varepsilon/4$ on each $i \in [n]$ and output the set of indices i for which we had large estimate of x_i, i.e. at least $(3\varepsilon/4)\|x\|_1$ (pretend we know $\|x\|_1$ exactly, which is trivial to do in the strict turnstile model).

2.1 CountMin sketch

We here describe the CountMin sketch [CM05], which solves ℓ_1 point query in the general turnstile model. We will describe it here in the strict turnstile model. We now describe the operation of the algorithm:

1. We store hash functions $h_1, ...h_L : [n] \to [t]$, each chosen independently from a 2-wise independent family.
2. We store counters $C_{a,b}$ for $a \in [s]$, $b \in [t]$ with $s = \lceil 2/\varepsilon \rceil$, $t = \lceil \log_2(1/\delta) \rceil$.
3. Upon an update (i, Δ), we add Δ to all counters $C_{a,h_a(i)}$ for $a = 1, \ldots, s$.
4. To answer $query(i)$, we output $\min_{1 \le a \le s} C_{a,h_a(i)}$.

Note that our total memory consumption, in words is $m = O(st) = O(\varepsilon^{-1} \log(1/\delta))$.

Claim 1. *$query(i) = x_i \pm \varepsilon\|x\|_1$ w.p $\ge 1 - \delta$.*

Proof. Fix i, let $Z_j = 1$ if $h_r(j) = h_r(i)$, $Z_j = 0$ otherwise. Now note that for any $r \in [s]$, $C_{r,h_r(i)} = x_i + \underbrace{\sum_{j \neq i} x_j Z_j}_{E}$. We have $\mathbb{E}(E) = \sum_{j \neq i} |x_j| \mathbb{E} Z_j = \sum_{j \neq i} |x_j|/t \le \varepsilon/2 \cdot \|x\|_1$. Thus by Markov's inequality, $\mathbb{P}(E > \varepsilon\|x\|_1) < 1/2$. Thus by independence of the s rows of the CountMin sketch, $\mathbb{P}(\min_r C_{r,h_r(i)} > x_i + \varepsilon\|x\|_1) < 1/2^L = \delta$. □

Thus we easily obtain the following theorem.

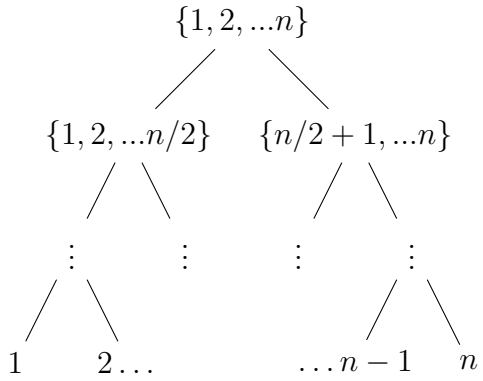
Theorem 1. *There is an algorithm solving the ℓ_1 ε-heavy hitter problem in the strict turnstile model with failure probability δ, space $O(\varepsilon^{-1}\log(n/\delta))$, update time $O(\log(n/\delta))$, and query time $O(n\log(n/\delta))$.*

Proof. We can instantiate a point query data structure with failure probability δ/n. Then we point query every $i \in [n]$ and include in our outpust list L only those i for which query returned a value at least $(3\varepsilon/4)\|x\|_1$. □

2.1.1 Speeding up query time

While the above theorem gives *some* correct heavy hitter algorithm with small space, the quer ytime is quite slow. Here we discuss the *dyadic trick* technique of [CM05] to speed up query.

Consider a perfect binary tree whose leaves are in correpondence with $[n]$.



There are $1 + \lg n$ levels of the tree. We imagine level j of the tree (with the root being level 0) corresponds to a 2^j-dimensional vector $x(j)$ being updated. Each node in the tree is a coordinate of the vector at the corresponding level, and the value at that coordinate is the sum of the two values of the children (with ith leaf simply having value x_i). Then when we see an update (i, Δ), we imagine that this update happens to all coordinates that are ancestors of the ith leaf. Formally, what we actually store in memory is $1 + \lg n$ CM sketches, one per level. Then upon an update, we feed that update to the appropriate coordinate at the CM sketch at every level.

We henceforth imagine we are trying to solve the α-heavy hitters problem for some $0 < \alpha < 1$ (to distinguish from the ε in the point query problem). If our goal is to have final failure probability δ, then each CM sketch has error parameter $\varepsilon = \alpha/4$ and failure probabliity $\eta = \delta\alpha/(4\lg n)$.

To answer a query, the key insight is that in the strict turnstile model *the value at any ancestor of a node is at least as big as the value at that node*, and furthermore the ℓ_1 norm of the implicit vector at each level of the tree is exactly the same. Therefore, if i is a heavy hitter for the vector x at the lowest level of the tree, then *every* ancestor of i is a heavy hitter at its level as well. Since there can only be at most $2/\alpha$ indices that are $\alpha/2$ heavy hitters, this suggests the following depth first search tree. We move down the tree starting from the root (the root vertex is certainly a 1-heavy hitter for its 1-dimensional vector). At each level j of the tree, we keep track of a list L_j of heavy hitters at that level (L_j should contain all α-heavy hitters of the vector at its level, and no one below $\alpha/2$-heavy). Then, for each of the two children of an index in L_j, we point query that child using the CM sketch at level $j+1$. If a child has point query output at least $(3\alpha/4)\|x\|_1$, we include it in L_{j+1}. Finally, our final output list L is simply the list corresponding to the bottom-most level of the tree.

Correctness. Conditioned on every CountMin sketch at every level being correct, each L_j can have size at most $2/\alpha$, and thus on level $j+1$ we point query at most $4/\alpha$ nodes (if this is ever not the case and we find ourselves querying more, we can simply output Fail). Thus since we only do at most $Q \leq (4 \lg n)/\alpha$ queries, since each CM sketch has failure probability at most δ/Q, by a union bound our total failure probability is at most δ.

Complexity. The space used is $O(\varepsilon^{-1} \lg n \lg(1/\eta)) = O(\varepsilon^{-1} \lg n \lg((\lg n)/(\alpha\delta)))$ (in words). The query time is the same. The update time is $O(\lg n \lg(1/\eta)) = O(\lg n \lg((\lg n)/(\alpha\delta)))$.

Though we will not discuss it here, currently the best known algorithm for ℓ_1 heavy hitters in the turnstile model is the ExpanderSketch of [LNNT16]. It achieves $O(\varepsilon^{-1} \lg(n/\delta))$ words of space, $O(\lg(n/\delta))$ update time, and $O(\varepsilon^{-1} \lg(n/\delta)\, poly(\lg n))$ query time to achieve failure probability δ.

2.1.2 ℓ_1/ℓ_1 sparse recovery

Here we show that the CountMin sketch can be used to solve the ℓ_1/ℓ_1 sparse recovery problem. First, without justification, we claim the CountMin sketch solves point query with the same complexity even in the general turnstile model (when some x_i can be negative). The only difference in the algorithm

is that we replace the min estimator with a median and increase s by a constant factor (the analysis uses the Chernoff bound).

In the ℓ_1/ℓ_1 recovery problem, we wish to recovery a vector y such that $\|x - y\|_1 \leq (1+\varepsilon)\|x_{tail(k)}\|_1$. The question is can you get to l_1/l_1 for HH. CM sketch can give this with $\|y\|_0 \leq k$

Definition 1. *$x_{tail(k)}$ is x but with the heaviest k coordinates in magnitude zeroed out. Thus for example, note that if x is actually k-sparse then we recover x exactly with zero error.*

When using the CM sketch, let us define x' to be the vector with $x'_i = query(i)$. We leave the proof of the following claim to you as an exercise, which is a slight strengthening of the point query error guarantee of the CM sketch.

Claim 2. *If CM has $t \geq \Theta(k/\varepsilon)$, $s = \Theta(\lg(1/\delta))$ then w.p. $1-\delta$, $x'_i = x_i \pm (\varepsilon/k)\|x_{tail(k)}\|_1$*

Let $T \subset [n]$ correspond to largest k entries of x' in magnitude. Now, we will return $y = x'_T$, i.e. the projection of x' onto coordinates in T.

Claim 3. $\|x - y\|_1 \leq (1+3\varepsilon)\|x_{tail(k)}\|_1$

Proof. Let S denote $head(x) \subset [n]$ and T denote $head(x') \subset [n]$. We have

$$\begin{aligned}
\|x-y\|_1 &= \|x\|_1 - \|x_T\|_1 + \|x_T - y_T\|_1 \\
&\leq \|x\|_1 + \|x_T - y_T + y_T\|_1 + \|x_T - y_T\|_1 \\
&\leq \|x\|_1 - \|y_T\|_1 + 2\|x_T - y_T\|_1 \\
&\leq \|x\| - \|y_S\| + 2\|x_T - y_T\|_1 \\
&\leq \|x\| - \|x_S\| + \|x_S - y_S\|_1 + 2\|x_T - y_T\|_1 \\
&\leq \|x_{tail(k)}\|_1 + 3\varepsilon\|x_{tail(k)}\|_1
\end{aligned}$$

□

2.2 Deterministic ℓ_1 point query and heavy hitters

Here we give an example application of the Johnson-Lindenstrauss (JL) lemma to heavy hitters.

Lemma 1 (Johnson-Lindenstrauss (JL) lemma [JL84]). *For any $X = \{x_1, \ldots, x_N\} \subset \mathbb{R}^n$ and $0 < \varepsilon < 1/2$, there exists $f : X \to \mathbb{R}^m$ for $m = O(\varepsilon^{-2} \log N)$ such that for all $1 \le i < j \le N$,*

$$(1-\varepsilon)\|x_i - x_j\|_2 \le \|f(x_i) - f(x_j)\|_2 \le (1+\varepsilon)\|x_i - x_j\|_2. \tag{1}$$

Furthermore, the map f can be taken to be linear: $f(x) = \Pi x$ for some $\Pi \in \mathbb{R}^{m \times n}$.

Recall for ℓ_1 point query, the query receives an index $i \in [n]$, and the response to the query should be a value $\tilde{x}$ such that $|x_i - \tilde{x}_i| \le \varepsilon \|x\|_1$. We show an argument of [NNW14] that the JL lemma implies the existence of a fixed deterministic $\Pi \in \mathbb{R}^{m \times n}$ with $m \lesssim \varepsilon^{-2} \log n$ such that such a $\tilde{x}$ can be recovered from Πx.

Definition 2. *We say that a matrix Π with columns $\Pi_1, \ldots, \Pi_n$ is* ε-incoherent *if (1) $\|\Pi_i\|_2 = 1$ for all i, and (2) for all $i \ne j$, $|\langle \Pi_i, \Pi_j \rangle| \le \varepsilon$.*

Theorem 2. *If $\Pi \in \mathbb{R}^{m \times n}$ is ε-incoherent, then there is a polynomial time recovery algorithm $\mathcal{A}_\Pi$ such that given any $y = \Pi x$, if we define $\tilde{x} = \mathcal{A}_\Pi(y)$ then $\|\tilde{x} - x\|_\infty \le \varepsilon \|x\|_1$.*

Proof. The recovery algorithm will be $\mathcal{A}_\Pi(y) = \Pi^T y = \Pi^T \Pi x$. Thus

$$\tilde{x}_i = e_i^T \Pi^T \Pi x = \sum_{j=1}^{n} \langle \Pi_i, \Pi_j \rangle x_i = x_i + \sum_{i \ne j} \langle \Pi_i, \Pi_j \rangle x_i = x_i \pm \varepsilon \|x\|_1.$$

□

Now we show the existence of such Π with small m.

Lemma 2. *$\forall\ \varepsilon \in (0, 1/2)$, there is ε-incoherent Π with $m \lesssim \varepsilon^{-2} \log n$.*

Proof. Consider the set of vectors $\{0, e_1, \ldots, e_n\}$. By the JL lemma, there exists Π' with $O(\varepsilon^{-2} \log n)$ rows, and having columns Π'_i such that (1) $|\Pi'_i\|_2 = \|\Pi' e_i\|_2 = 1 \pm \varepsilon/3$, and (2) $\|\Pi'_i - \Pi'_j\|_2 = \|\Pi' e_i - \Pi' e_j\|_2 = (1 \pm \varepsilon/3)\sqrt{2}$ for all $i \ne j$. Let Π be the matrix whose ith column is $\Pi'_i / \|\Pi'_i\|_2$. Then $\|\Pi_i\|_2 = 1$ for all i, as desired. Furthermore

$$2(1 \pm \varepsilon)^2 = \|\Pi_i - \Pi_j\|_2^2 = \|\Pi_i\|_2^2 + \|\Pi_j\|_2^2 - 2\langle \Pi_i, \Pi_j \rangle.$$

Note $\|\Pi_i\|_2^2$ and $\|\Pi_j\|_2^2$ are both $1 \pm O(\varepsilon)$, implying $|\langle \Pi_i, \Pi_j \rangle| = O(\varepsilon)$. The lemma follows by applying this argument with ε scaled down by a constant. □

References

[AHLW16] Yuqing Ai, Wei Hu, Yi Li, and David P. Woodruff. New characterizations in turnstile streams with applications. In *Proceedings of the 31st Conference on Computational Complexity (CCC)*, pages 20:1–20:22, 2016.

[AMS99] Noga Alon, Yossi Matias, and Mario Szegedy. The space complexity of approximating the frequency moments. *J. Comput. Syst. Sci.*, 58(1):137–147, 1999.

[BJKS04] Ziv Bar-Yossef, T. S. Jayram, Ravi Kumar, and D. Sivakumar. An information statistics approach to data stream and communication complexity. *J. Comput. Syst. Sci.*, 68(4):702–732, 2004.

[CM05] Graham Cormode and S. Muthukrishnan. An improved data stream summary: the count-min sketch and its applications. *J. Algorithms*, 55(1):58–75, 2005.

[Ind06] Piotr Indyk. Stable distributions, pseudorandom generators, embeddings, and data stream computation. *J. ACM*, 53(3):307–323, 2006.

[IW05] Piotr Indyk and David P. Woodruff. Optimal approximations of the frequency moments of data streams. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing (STOC)*, pages 202–208, 2005.

[JL84] William B. Johnson and Joram Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. *Contemporary Mathematics*, 26:189–206, 1984.

[LNNT16] Kasper Green Larsen, Jelani Nelson, Huy L. Nguyễn, and Mikkel Thorup. Heavy hitters via cluster-preserving clustering. In *Proceedings of the 57th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2016.

[LNW14] Yi Li, Huy L. Nguyen, and David P. Woodruff. Turnstile streaming algorithms might as well be linear sketches. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing (STOC)*, pages 174–183, 2014.

[NNW14] Jelani Nelson, Huy L. Nguyễn, and David P. Woodruff. On deterministic sketching and streaming for sparse recovery and norm estimation. *Linear Algebra and its Applications, Special Issue on Sparse Approximate Solution of Linear Systems*, 441:152–167, 2014.