The turnstile streaming model / Linear Sketching
Sparse recovery

Def. A data structure $\text{DS}(\Omega)$ computed from stream $\Omega$ is a \underline{sketch} if there is a combining alg
\[ \text{Comb} (\text{DS}(\Omega_1), \text{DS}(\Omega_2)) = \text{DS}(\Omega_1 \circ \Omega_2) \]
that can be computed in small space.

A sketching alg is a linear sketch if
\[ \text{sk}(\Omega) \in \text{vector space of dim } \ell = l(n) \]
A x implicitly & \[ \text{sk}(\Omega) \] is a linear function the
frequency vector \((f_1, f_2, \ldots, f_n)\), where
\[ f_i = \# \text{ occurrences of elt } i \text{ in } \Omega. \]
Turnstile token \((i, c)\)

\[ \text{update for } \text{elt} \in \text{set}\]

\[ f_j \leftarrow f_j + c \]

\[ c \in \mathbb{Z} \]

Streaming \(s\)-sparse recovery / detection.

if \(T\) has \(\leq s\) values

\[ \implies f \text{ is } \text{sparse}, |\text{supp}(f)| \leq s \]

output \(f\) (we only care about the \(s\)-elements of \(\text{supp}\))

otherwise output "not \(s\)-sparse"
freq. vec. of stream

\( f: 0 \quad 0 \quad 2 \quad 4 \quad 0 \quad 0 \quad 1 \quad 0 \)

\( f_2 \uparrow \uparrow \quad \uparrow \quad \uparrow \quad \uparrow \quad \uparrow \quad \uparrow \quad \uparrow \quad \uparrow \)

\((3,2) \quad (4,4) \quad (7,1)\)

\( \text{post count/freq. of } i \quad \forall \ i \in \text{supp}(f) \)

\( O(s \log n) \) space needed.

Eg:

updates

\( (2, 3) \quad f_2 = 0 \)

\( (2, -3) \quad f_2 = 3 - 3 = 0 \)

\( (2, 1) \quad f_2 = 1 \)

\( (2, -2) \quad f_3 = -1 \)

Notation:

sparsity \( S = \ell_0(f) \)

Space:

\( \text{poly}(S, \log n) \); \quad S < n \)
Special case $s = 1$

$$f = (0 \ 0 \ 0 \ \lambda \ \ 0 \ 0 \ 0)$$

$$f_i = \lambda.$$

Promiss: $f$ is 1-sparse.

Alg: maintain

$$l = \sum f_i \leftarrow \text{sum of freq. lin sketches.}$$

$$z = \sum j f_j \leftarrow \text{sum of all of these.}$$

if $1$-sparsity

$$l = k$$

$$z = \lambda i \quad \Rightarrow \\ i = \frac{z}{\lambda}$$

Output $\frac{z}{l}$ if $l \neq 0$

$$\theta^m.$$
No promiss case:

Use "fingerprint": map [big] \to [small]

\Pr[\text{collision}] \text{ is small}

Let $F = \mathbb{F}_p$ ($\mathbb{F}_p^n \to \mathbb{F}_p^n$)

Idea: maintain $p(x) = \sum_{i=1}^{n} f_i x^i$ implicitly

Alg for 1-sparse recovery & detection

- Initialize $(\ell, z, P)$

- Pick $r \in F^*$ \rightarrow $Z \mod p$ prime $p \in \Theta(n^3)$

- On token $(j, c)$

  \begin{align*}
  l &= l + c \\
  z &= z + j \cdot c \\
  p &= p + c \cdot \text{fingerprint}
  \end{align*}

  \text{update } f_j \leftarrow \text{fingerprint} \\
  j \leq n \\
  \text{updates just the coeff at } p_j$
\[
\begin{align*}
\text{Output:} & \quad p(x) = 2^{f_{x\text{fix}}}, \quad r \text{ is maintained in } \text{simplicity} \\
\text{coefficients: frequencies} & \\
\text{if } \ell = 2 = p_2 = 0 \text{ out } f = 0 & \Rightarrow f = 0 \quad \ell' = 2/2, \ell = 2 \\
\text{else } f = 2/2 \text{ or } f = 2/1 & \Rightarrow f = 2/2 \text{ or } f = 2/1 \\
P & = 2^{r - 3} - 3^r \\
\text{sequence of updates} & \\
P & = 2^{r - 3} - 3^r \\
\text{size of domain} & = 2^{2r + 2} \\
P & = 2^{2r + 2} \\
\text{output} & = \log_2 m \\
P & = \log_2 m \\
\text{else output} & \\
f & \geq 1/2 \ell, \ell' \\
P & = \log_2 m \\
\text{else output} & \\
f & \geq 1/2 \ell, \ell' \\
P & = \log_2 m \\
\text{else output} & \\
f & \geq 1/2 \ell, \ell' \\
P & = \log_2 m \\
\text{else output} & \\
f & \geq 1/2 \ell, \ell' \\
P & = \log_2 m \\
\text{else output} & \\
f & \geq 1/2 \ell, \ell' \\
P & = \log_2 m
\end{align*}
\]
Analysis:

If $f$ is 1-sparse

then $L, Z, p$ are in the correct form.

$f$ will not output $Lz_0$ or $Zz_0$

nor $p_0$

so won’t output $f_0 = 0$

Also $\frac{z}{l} \in \mathbb{Z} \forall$

If $f$ is $>1$-sparse

$\Pr \left[ \text{Outputting incorrect "1-sparse" answer} \right]

= \Pr \left[ p(r) = 0 \frac{z}{l} \right]

\text{w.r.t.}

p(x) \neq ln^{r/l} \text{ as polynomials}
\( p \) is at least 2-sparse

\[
\begin{bmatrix}
0 & 0 & 1 & 1 & 0 & 0 & 0
\end{bmatrix}
\neq
\begin{bmatrix}
0 & 0 & 0 & 0 & 1 & 0 & 0
\end{bmatrix}
\]

\[ x^3 + x^4 \]

\[ \Pr \left[ \left( p - x^{2/e} \right) (r) = 0 \right] \]

\[ \text{poly of deg } \leq n \]

\[ \frac{n}{|F|} \leq \frac{1}{n^2} \]
General s-sparse recovery s defection

2 10 + 2 n \text{---} j

Use hashing \( h : [n] \rightarrow [s'] \)

\[
\begin{array}{cccccc}
1 & 2 & 3 & \text{---} & n \\
\hline
\text{bucket 1} & & & & \text{bucket 2} \\
\end{array}
\]

\( s' \ll n \)

We'll use a stream per bucket (in parallel)

Point: If stream s-sparse then two distinct elements of \( \text{supp}(f) \) are hashed into different buckets

So we'll be using Alg for 1-sparse recovery.
E.g.: \( \text{Supp}(f) = \{3, 4, 7, n\} \)

\[ h: [n] \rightarrow [6] \quad \text{from 2-universal family} \]

Use 2-universal hash family

\[ H = \{ h : [n] \rightarrow [2S] \} \quad \text{is 2-universal} \]

\( \forall x \neq y \in [n] \)

\[ \Pr \left[ h(x) = h(y) \right] \leq \frac{1}{2S} \]

Goal: Run many 1-sparse recovery instances in parallel \((2S \text{ per hash function})\)
Arg for $\delta$-sparse recovery:

Maintain

$t = O(\log s)$

\[ D[t][2s] = 0 \]

On token $(j, c)$

Look at where $j$ is mapped to by each $h_i$. For each hash $i$ update $D[i][h_i(j)]$.

This stream should take token $(h_i(j), c)$ in $\delta$-sparse Subroutine.
(as if D[i][h(i, s)] datastructure is for 1-sparse streams)

Output:

For h: h ≠ k

if D[i][h] outputs 1-sparse

then

record it in A[i] = λi (overwrite if needed)

If |supp A| > s output "not s-sparse"

Otherwise output

\[ f = \sum_{i \in \text{supp}(A)} \lambda_i \]

Analysis:

Alg is correct if:

1. If i ∈ supp(f)

   \( \exists \) hash that hashes \( i \) in its own bucket (without other snippets)

2. Otherwise output is a False Positive.
For fixed $i \in \text{Supp}(f)$, for the $j$th hash func.

\[
\Pr\left[ h_j(i) = h_j(k) \text{ for some other } k \in \text{Supp}(f) \right] \leq \sum_{k \neq i} \Pr\left[ h_j(i) = h_j(k) \right] \\
\leq \frac{s-1}{2s} < \frac{1}{2}
\]

by 2-wise min. of Hash family.

For fixed $i$

\[
\Pr\left[ \text{all } t \text{ of the hash funcs cause some collision in the supp} \right] \leq \left( \frac{1}{2} \right)^t \leq \frac{s}{5}
\]

for $t = \log \frac{s}{5}$

So by union bound

\[
\Pr\left[ \text{some } i \text{ in the supp is failed by all hashes} \right] \leq s \cdot \frac{s}{5} = \frac{s^2}{5}
\]
So every $i \in \text{Supp}(f)$ is split by some hash $j$ w.p. $1-\delta$.

So we succeed in getting the correct $1$-sparse streams w.p $1-\delta$.

So some non $1$-sparse stream outputs a false positive $1$-sparse answer $(i, x_i)$.

By union bound,

$$\leq 25 \cdot \epsilon \cdot \frac{1}{n^2} = o(1)$$

So we get correct answer w.p.

$$1-\delta - o(1)$$
Can do $s$-sparse recovery in the turnstile model w.p. $1 - \delta$ using $O(s \log \frac{s}{\delta} \log n)$ bits of space.

Can be viewed as maintaining

$$A \times \sum_{i \in \text{freq}} \text{random matrix}$$

$A$-sparse rec. : $(f_1, f_2, \ldots, f_n)$

$$\begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{pmatrix}$$

$$= \sum f_i r_i$$

$s$-sparse : take multiple random $r$'s.
implicitly maintaining a Vandermonde matrix (random)

linear sketch: $V \cdot f$ is maintained implicitly in small space.
One can use same ideas as above to get "lo-sampling"

lo-sampling: sample a uniform random elt in the supp \( f \)

\[ \forall i \in \text{supp} \quad \Pr[ \text{output} = i ] = \frac{1}{|\text{supp}(f)|} \]

Thm: Jayaram & Woodruff.

Can do perfect lo-sampling in small space using linear sketch,

\[ \Pr[ \text{sketch } A \text{ gives correct answer } ] \geq 1 - \delta \]

\( A \)
Eg: Sample a unit random neighbor of a \( u \times v \) vector of \( u \):

\[
\begin{pmatrix}
0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0
\end{pmatrix}
\]

\( l_0 \) sampling: output unit random set of support \( \text{adj}(v) \)

We'll use \( l_0 \)-sampling as a subroutine in graph sketching for connectivity & bipartiteness testing.