Applications of CountMin and Count Sketches

Lecture 10
September 24, 2020
CountMin Sketch

\textbf{CountMin-Sketch}(w, d):
\begin{itemize}
  \item \(h_1, h_2, \ldots, h_d\) are pair-wise independent hash functions from \([n] \rightarrow [w]\).
  \item While (stream is not empty) do
    \begin{itemize}
      \item \(e_t = (i_t, \Delta_t)\) is current item
      \item for \(\ell = 1\) to \(d\) do
        \begin{itemize}
          \item \(C[\ell, h_\ell(i_j)] \leftarrow C[\ell, h_\ell(i_j)] + \Delta_t\)
        \end{itemize}
    \end{itemize}
  \end{itemize}
\end{itemize}

For \(i \in [n]\) set \(\tilde{x}_i = \min_{\ell = 1}^{d} C[\ell, h_\ell(i)]\).

Counter \(C[\ell, j]\) simply counts the sum of all \(x_i\) such that \(h_\ell(i) = j\). That is,
\[C[\ell, j] = \sum_{i : h_\ell(i) = j} x_i.\]
Summarizing

Lemma

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

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

Corollary

With \( d = \Omega(\ln n) \) and \( w = 2/\epsilon \), with probability \( (1 - \frac{1}{n}) \) for all \( i \in [n] \):

\[
\tilde{x}_i \leq x_i + \epsilon \|x\|_1.
\]

Total space: \( O\left(\frac{1}{\epsilon} \log n\right) \) counters and hence \( O\left(\frac{1}{\epsilon} \log n \log m\right) \) bits.
Count Sketch

Count-Sketch\((w, d)\):

- \(h_1, h_2, \ldots, h_d\) are pair-wise independent hash functions from \([n] \rightarrow [w]\).
- \(g_1, g_2, \ldots, g_d\) are pair-wise independent hash functions from \([n] \rightarrow \{-1, 1\}\).

While (stream is not empty) do

- \(e_t = (i_t, \Delta_t)\) is current item
- for \(\ell = 1\) to \(d\) do

  \[C[\ell, h_\ell(i)] \leftarrow C[\ell, h_\ell(i)] + g(i_t)\Delta_t\]

endWhile

For \(i \in [n]\)

set \(\tilde{x}_i = \text{median}\{g_1(i)C[1, h_1(i)], \ldots, g_\ell(i)C[\ell, h_\ell(i)]\}\).
Lemma

Let $d \geq 4 \log \frac{1}{\delta}$ and $w > \frac{3}{\epsilon^2}$. Then for any fixed $i \in [n]$, $E[\tilde{x}_i] = x_i$ and $\Pr[|\tilde{x}_i - x_i| \geq \epsilon \|x\|_2] \leq \delta$.

Corollary

With $d = \Omega(\ln n)$ and $w = 3/\epsilon^2$, with probability $(1 - \frac{1}{n})$ for all $i \in [n]$: $|\tilde{x}_i - x_i| \leq \epsilon \|x\|_2$.

Total space $O(\frac{1}{\epsilon^2} \log n)$ counters and hence $O(\frac{1}{\epsilon^2} \log n \log m)$ bits.
Part I

Applications
Heavy Hitters: Point queries

Heavy Hitters Problem: Find all items $i$ such that $x_i > \alpha \|x\|_1$ for some fixed $\alpha \in (0, 1]$.

Approximate version: output any $i$ such that $x_i \geq (\alpha - \epsilon) \|x\|_1$

The sketches give us a data structure such that for any $i \in [n]$ we get an estimate $\tilde{x}_i$ of $x_i$ with additive error.
Heavy Hitters: Point queries

**Heavy Hitters Problem:** Find all items \( i \) such that \( x_i > \alpha \| x \|_1 \) for some fixed \( \alpha \in (0, 1] \).

Approximate version: output any \( i \) such that \( x_i \geq (\alpha - \epsilon) \| x \|_1 \)

The sketches give us a data structure such that for any \( i \in [n] \) we get an estimate \( \tilde{x}_i \) of \( x_i \) with additive error.

Go over each \( i \) and check if \( \tilde{x}_i > (\alpha - \epsilon) \| x \|_1 \).
Heavy Hitters: Point queries

**Heavy Hitters Problem:** Find all items $i$ such that $x_i > \alpha \|x\|_1$ for some fixed $\alpha \in (0, 1]$.

Approximate version: output any $i$ such that $x_i \geq (\alpha - \epsilon) \|x\|_1$

The sketches give us a data structure such that for any $i \in [n]$ we get an estimate $\tilde{x}_i$ of $x_i$ with additive error.

Go over each $i$ and check if $\tilde{x}_i > (\alpha - \epsilon) \|x\|_1$. Expensive
Heavy Hitters: Point queries

**Heavy Hitters Problem:** Find all items $i$ such that $x_i > \alpha \|x\|_1$ for some fixed $\alpha \in (0, 1]$.

Approximate version: output any $i$ such that $x_i \geq (\alpha - \epsilon) \|x\|_1$

The sketches give us a data structure such that for any $i \in [n]$ we get an estimate $\tilde{x}_i$ of $x_i$ with additive error.

Go over each $i$ and check if $\tilde{x}_i > (\alpha - \epsilon) \|x\|_1$. Expensive

Additional data structures to speed up above computation and reduce time/space to be proportional to $O\left(\frac{1}{\alpha} \text{polylog}(n)\right)$. More tricky for Count Sketch. See notes and references.
Range Queries

Range query: given $i, j \in [n]$ want to know $\sum_{i \leq l \leq j} x[i, j]$

Examples:

- $[n]$ corresponds to IP address space in network routing and $[i, j]$ corresponds to addresses in a range
- $[n]$ corresponds to some numerical attribute in a database and we want to know number of records within a range
- $[n]$ corresponds to the discretization of a signal value
\[(0, 1, 10, 5, 3, 2), 10, \ldots, \sqrt{i} \div \frac{1}{j} \]
Range Queries

Range query: given $i, j \in [n]$ want to know $\sum_{i \leq l \leq j} x[i, j]$

Examples:

- $[n]$ corresponds to IP address space in network routing and $[i, j]$ corresponds to addresses in a range
- $[n]$ corresponds to some numerical attribute in a database and we want to know number of records within a range
- $[n]$ corresponds to the discretization of a signal value

Want to create a sketch data structure that can answer range queries for any given range that is chosen after the sketch is done. $\Omega(n^2)$ potential queries
Range Queries

Simple idea: imagine a binary tree over $[n]$ and any interval $[i, j]$ can be broken up into $O(\log n)$ disjoint "dyadic" intervals.

\[ (1, 2, \ldots, n) \]

\[ n = 2^\Theta \]

To manage error choose $\varepsilon_0 = \varepsilon / \log n$: total space is $O(\varepsilon \log n / \log \varepsilon)$ where $\varepsilon$ is the space for single level sketch.

\[ \sum_{l=0}^{\log n} x_{ij} \]

\[ (x_1, x_2, \ldots) \]

\[ (\tilde{x}_1, \tilde{x}_2, \ldots) \]
Range Queries

**Simple idea:** imagine a binary tree over \([n]\) and any interval \([i, j]\) can be broken up into \(O(\log n)\) disjoint "dyadic" intervals

Create one sketch data structure per level of binary tree
Range Queries

**Simple idea:** imagine a binary tree over \([n]\) and any interval \([i, j]\)
can be broken up into \(O(\log n)\) disjoint "dyadic" intervals.

Create one sketch data structure per level of binary tree.

Output estimate \(\tilde{x}[i,j]\) by adding estimates for \(O(\log n)\) dyadic
intervals that \([i,j]\) decomposes into.
Range Queries

**Simple idea:** imagine a binary tree over $[n]$ and any interval $[i, j]$ can be broken up into $O(\log n)$ disjoint ”dyadic” intervals

Create one sketch data structure per level of binary tree

Output estimate $\tilde{x}[i, j]$ by adding estimates for $O(\log n)$ dyadic intervals that $[i, j]$ decomposes into

To manage error choose $\epsilon' = \epsilon / \log n$: total space is $O(\alpha \log n / \epsilon)$ where $\alpha$ is the space for single level sketch
Part II

Sparse Recovery
**Sparse Recovery**

**Sparsity** is an important theme in optimization/algorithms/modeling.

- Data is often *explicitly* sparse. Examples: graphs, matrices, vectors, documents (as word vectors).
- Data is often *implicitly* sparse — in a different representation the data is explicitly sparse. Examples: signals/images, topics, etc.

\[ \mathbf{x} = (x_1, x_2, \ldots, x_T) \]
Sparse Recovery

Sparsity is an important theme in optimization/algorithms/modeling:

- Data is often *explicitly* sparse. Examples: graphs, matrices, vectors, documents (as word vectors)
- Data is often *implicitly* sparse — in a different representation the data is explicitly sparse. Examples: signals/images, topics, etc

Algorithmic goals:

- Take advantage of sparsity to improve performance (speed, quality, memory etc)
- Find implicit sparse representation to reveal information about data. Example: topics in documents, frequencies in Fourier analysis
Sparse Recovery

**Problem:** Given vector/signal $x \in \mathbb{R}^n$ find a sparse vector $z$ such that $z$ approximates $x$

**More concretely:** given $x$ and integer $k \geq 1$, find $z$ such that $z$ has at most $k$ non-zeroes ($\|z\|_0 \leq k$) such that $\|x - z\|_p$ is minimized for some $p \geq 1$.

**Optimum offline solution:** $z$ picks the largest $k$ coordinates of $x$ (in absolute value)

Want to do it in streaming setting: turnstile streams and $p = 2$ and want to use $\tilde{O}(k)$ space proportional to output

$(0, 10, 1, 9, 0.1, -0.2, 2, 0.001, 5)$

$(0, 10, 0, 9, 0, 0, 0, 0, 5)$
Sparse Recovery under $\ell_2$ norm

Formal objective function:

$$\alpha_k = \text{err}_2^k(x) = \min_{z: \|z\|_0 \leq k} \|x - z\|_2$$

For instance when $x$ is uniform, say $x_i = 1$ for all $i$ then $\|x\|_2^2 = \sqrt{n}$ but $\text{err}_2^k(x) = \sqrt{n}$ and hence related to distinct element detection.
Sparse Recovery under $\ell_2$ norm

Formal objective function:

\[
\text{err}_2^k(x) = \min_{z: \|z\|_0 \leq k} \|x - z\|_2
\]

$\text{err}_2^k(x)$ is interesting only when it is small compared to $\|x\|_2$

For instance when $x$ is uniform, say $x_i = 1$ for all $i$ then

$\|x\|_2 = \sqrt{n}$ but $\text{err}_2^k(x) = \sqrt{n - k}$

$\text{err}_2^k(x) = 0$ iff $\|x\|_0 \leq k$ and hence related to distinct element detection

\[
\begin{pmatrix}
1, 1, 1, \ldots, 1 \\
1, 0, 0, 0, 0
\end{pmatrix}
\]

$\|x\|_2 = \sqrt{n}$

$\|x-H\|_2 = \sqrt{n-k}$. 
Sparse Recovery under $\ell_2$ norm

**Theorem**

There is a linear sketch with size $O\left(\frac{k}{\epsilon^2} \text{polylog}(n)\right)$ that returns $z$ such that $\|z\|_0 \leq k$ and with high probability $\|x - z\|_2 \leq (1 + \epsilon) \text{err}_2^k(x)$.

Hence space is proportional to desired output. Assumption $k$ is typically quite small compared to $n$, the dimension of $x$.

Note that if $x$ is $k$-sparse vector is exactly reconstructed

Based on CountSketch
Algorithm

- Use Count Sketch with $w = \frac{3k}{\varepsilon^2}$ and $d = \Omega(\log n)$.
- Count Sketch gives estimates $\tilde{x}_i$ for each $i \in n$.
- Output the $k$ coordinates with the largest estimates.
Algorithm

- Use Count Sketch with $w = \frac{3k}{\epsilon^2}$ and $d = \Omega(\log n)$.
- Count Sketch gives estimates $\tilde{x}_i$ for each $i \in n$
- Output the $k$ coordinates with the largest estimates

$\left| \tilde{x}_i - x_i \right| \leq \epsilon \cdot \|x\|_2 \cdot \kappa_k$

Intuition for analysis

- With $w = \frac{ck}{\epsilon^2}$ the $k$ biggest coordinates will be spread out in their own buckets
- rest of small coordinates will be spread out evenly
- refine the analysis of Count-Sketch to carefully analyze the two scenarios
Lemma

Count-Sketch with $w = 3k/\varepsilon^2$ and $d = O(\log n)$ ensures that

$$\forall i \in [n], \quad |\tilde{x}_i - x_i| \leq \frac{\varepsilon}{\sqrt{k}} \text{err}_2^k(x)$$

with high probability (at least $(1 - 1/n)$).

Lemma

Let $x, y \in \mathbb{R}^n$ such that $\|x - y\|_\infty \leq \frac{\varepsilon}{\sqrt{k}} \text{err}_2^k(x)$. Then,

$$\|x - z\|_2 \leq (1 + 5\varepsilon) \text{err}_2^k(x),$$

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

Lemmas combined prove the correctness of algorithm.
Count Sketch

Count-Sketch\((w, d)\):

- \(h_1, h_2, \ldots, h_d\) are pair-wise independent hash functions from \([n] \rightarrow [w]\).
- \(g_1, g_2, \ldots, g_d\) are pair-wise independent hash functions from \([n] \rightarrow \{-1, 1\}\).

While (stream is not empty) do

\(e_t = (i_t, \Delta_t)\) is current item

for \(\ell = 1\) to \(d\) do

\[C[\ell, h_\ell(i_t)] \leftarrow C[\ell, h_\ell(i_t)] + g(i_t)\Delta_t\]

endWhile

For \(i \in [n]\)

set \(\hat{x}_i = \text{median}\{g_1(i)C[1, h_1(i)], \ldots, g_d(i)C[d, h_d(i)]\}\).
Recap of Analysis

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 $h_\ell$. $E[Y_{i'}] = E[Y_{i'}^2] = 1/w$ from pairwise independence of $h_\ell$.

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

Therefore,

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

because $E[g_\ell(i) g_\ell(i')] = 0$ for $i \neq i'$ from pairwise independence of $g_\ell$ and $Y_{i'}$ is independent of $g_\ell(i)$ and $g_\ell(i')$. 

Chandra (UIUC)
Recap of Analysis

\[ Z_\ell = g_\ell(i) C[\ell, h_\ell(i)] \]. And \( E[Z_\ell] = x_i \).

\[
Var(Z_\ell) = E\left[ (Z_\ell - x_i)^2 \right]
\]
\[
= E \left[ \left( \sum_{i' \neq i} g_\ell(i) g_\ell(i') Y_{i'} x_{i'} \right)^2 \right]
\]
\[
= E \left[ \sum_{i' \neq i} x_{i'}^2 Y_{i'}^2 + \sum_{i' \neq i''} x_{i'} x_{i''} g_\ell(i') g_\ell(i'') Y_{i'} Y_{i''} x_{i'} x_{i''} \right]
\]
\[
= \sum_{i' \neq i} x_{i'}^2 E \left[ Y_{i'}^2 \right]
\]
\[
\leq \|x\|_2^2 / \nu.
\]
Refining Analysis

\[ T_{\text{big}} = \{ i' \mid i' \text{ is one of the } k \text{ biggest coordinates in } x \} \]

\[ T_{\text{small}} = [n] \setminus T \]

\[ \sum_{i' \in T_{\text{small}}} x_{i'}^2 = (\text{err}^k(x))^2 \]

\[ \ell_k \leq \frac{\ell}{\sqrt{k}} \leq \frac{\ell_k}{\sqrt{k}} \]

\[(0, 1, 100, -1, 0, \ldots)\]
Refining Analysis

\( T_{\text{big}} = \{ i' \mid i' \text{ is one of the } k \text{ biggest coordinates in } x \} \)

\( T_{\text{small}} = [n] \setminus T \)

\[ \sum_{i' \in T_{\text{small}}} x_{i'}^2 = (\text{err}_2^k(x))^2 \]

What is \( \text{Pr}\left[ |Z_\ell - x_i| \geq \frac{\epsilon}{\sqrt{k}} \text{err}_2^k(x) \right] \)?
Refining Analysis

\[ T_{\text{big}} = \{ i' \mid i' \text{ is one of the } k \text{ biggest coordinates in } x \} \]

\[ T_{\text{small}} = [n] \setminus T \]

\[ \sum_{i' \in T_{\text{small}}} x_{i'}^2 = (\text{err}_2^k(x))^2 \]

What is \( \Pr\left[ |Z_{\ell} - x_i| \geq \frac{\epsilon}{\sqrt{k}} \text{err}_2^k(x) \right] \) ?

**Lemma**

\[ \Pr\left[ |Z_{\ell} - x_i| \geq \frac{\epsilon}{\sqrt{k}} \text{err}_2^k(x) \right] \leq \frac{2}{5}. \]
Analysis

\[ Z_\ell = g_\ell(i) C[\ell, h_\ell(i)] \]

\[ \omega = \frac{3k}{\varepsilon^2} \]

Let \( A_\ell \) be event that \( h_\ell(i') = h_\ell(i) \) for some \( i' \in T_{\text{big}}, i' \neq i \)

**Lemma**

\[ \Pr[A_\ell] \leq \varepsilon^2 / 3. \text{ In other words with } 1 - \varepsilon^2 / 3 \text{ probability no big coordinates collide with } i \text{ under } h_\ell. \]
Analysis

\[ Z_\ell = g_\ell(i) C[\ell, h_\ell(i)]. \]

Let \( A_\ell \) be event that \( h_\ell(i') = h_\ell(i) \) for some \( i' \in T_{\text{big}}, i' \neq i \)

**Lemma**

\[ \Pr[A_\ell] \leq \epsilon^2/3. \text{ In other words with } 1 - \epsilon^2/3 \text{ probability no big coordinates collide with } i \text{ under } h_\ell. \]

- \( Y_{i'} \) indicator for \( i' \neq i \) colliding with \( i \).
  \[ \Pr[Y_{i'}] \leq 1/w \leq \epsilon^2/(3k). \]
- Let \( Y = \sum_{i' \in T_{\text{big}}} Y_{i'} \). \( \mathbb{E}[Y] \leq \epsilon^2/3 \) by linearity of expectation.
- Hence \( \Pr[A_\ell] = \Pr[Y \geq 1] \leq \epsilon^2/3 \) by Markov

Chandra (UIUC)  CS498ABD  21  Fall 2020  21 / 36
Analysis

\[ Z_\ell = g_\ell(i) C[\ell, h_\ell(i)] \]
\[ = x_i + \sum_{i' \in T_{\text{big}}} g_\ell(i) g_\ell(i') Y_{i'} x_{i'} + \sum_{i' \in T_{\text{small}}} g_\ell(i) g_\ell(i') Y_{i'} x_{i'} \]

Let \( Z'_\ell = \sum_{i' \in T_{\text{small}}} g_\ell(i) g_\ell(i') Y_{i'} \)

Lemma

\[ \Pr \left[ |Z'_\ell| \geq \frac{\epsilon}{\sqrt{k}} \text{err}_2^k(x) \right] \leq 1/3. \]

\[ \mathbb{E}[Z'_\ell] = 0 \]
\[ \mathbb{E}[|Z'_\ell|] \geq \frac{\epsilon}{\sqrt{k}} \text{err}_2^k(x) \]

\[ \epsilon \leq \|x\|_2 \]

\[ \|Z'_\ell\| \leq \frac{1}{3}. \]
Analysis

\[ Z_\ell = g_\ell(i)C[\ell, h_\ell(i)] \]
\[ = x_i + \sum_{i' \in T_{\text{big}}} g_\ell(i)g_\ell(i') Y_i' x_i' + \sum_{i' \in T_{\text{small}}} g_\ell(i)g_\ell(i') Y_i' x_i' \]

Let \( Z'_\ell = \sum_{i' \in T_{\text{small}}} g_\ell(i)g_\ell(i') Y_i' \)

Lemma

\[ \Pr\left[ |Z'_\ell| \geq \frac{\epsilon}{\sqrt{k}} \text{err}_2^k(x) \right] \leq 1/3. \]

- \( \mathbb{E}[Z'_\ell] = 0 \)
- \( \text{Var}(Z'_\ell) \leq \mathbb{E}\left[ (Z'_\ell)^2 \right] = \sum_{i' \in T_{\text{small}}} x_{i'}^2/w \leq \left( \frac{\epsilon^2}{3k} \right) (\text{err}_2^k(x))^2 \)
- By Chebyshev \( \Pr\left[ |Z'_\ell| \geq \frac{\epsilon}{\sqrt{k}} \text{err}_2^k(x) \right] \leq 1/3. \)
Analysis: Proof of lemma

Want to show:

**Lemma**

\[ \Pr \left[ |Z_\ell - x_i| \geq \frac{\epsilon}{\sqrt{k}} err^k_2(x) \right] \leq \frac{2}{5}. \]
**Analysis: Proof of lemma**

Want to show:

**Lemma**

\[
\Pr \left[ |Z_\ell - x_i| \geq \frac{\epsilon}{\sqrt{k}} \text{err}_2^k(x) \right] \leq 2/5.
\]

We have
\[
Z_\ell = g_\ell(i)C[\ell, h_\ell(i)] = x_i + \sum_{i' \in T_{\text{big}}} g_\ell(i)g_\ell(i') Y_{i'} x_{i'} + \sum_{i' \in T_{\text{small}}} g_\ell(i)g_\ell(i') Y_{i'} x_{i'}
\]
Analysis: Proof of lemma

Want to show:

**Lemma**

\[
\Pr\left[|Z_\ell - x_i| \geq \epsilon \frac{\text{err}_k}{\sqrt{k}}(x)\right] \leq 2/5.
\]

We have

\[
Z_\ell = g_\ell(i)C[\ell, h_\ell(i)] = x_i + \sum_{i' \in T_{\text{big}}} g_\ell(i) g_\ell(i') Y_{i'} x_{i'} + \sum_{i' \in T_{\text{small}}} g_\ell(i) g_\ell(i') Y_{i'} x_{i'}
\]

We saw:

**Lemma**

\[
\Pr\left[|Z_\ell'| \geq \epsilon \frac{\text{err}_k}{\sqrt{k}}(x)\right] \leq 1/3.
\]

**Lemma**

\[
\Pr[A_\ell] \leq \epsilon^2/3. \text{ In other words with } 1 - \epsilon^2/3 \text{ probability no big coordinates collide with } i \text{ under } h_\ell.
\]
Analysis: Proof of lemma

\[ Z_\ell = g_\ell(i) C[\ell, h_\ell(i)] \]
\[ = x_i + \sum_{i' \in T_{\text{big}}} g_\ell(i) g_\ell(i') Y_{i'} x_{i'} + \sum_{i' \in T_{\text{small}}} g_\ell(i) g_\ell(i') Y_{i'} x_{i'} \]

\[ |Z_\ell - x_i| \geq \frac{\epsilon}{\sqrt{k}} \text{err}_2^k(x) \text{ implies} \]

- \( A_\ell \) happens (that is some big coordinate collides with \( i \) in \( h_\ell \) or
- \( |Z'_\ell| \geq \frac{\epsilon}{\sqrt{k}} \text{err}_2^k(x) \)
Analysis: Proof of lemma

\[ Z_\ell = g_\ell(i)C[\ell, h_\ell(i)] \]
\[ = x_i + \sum_{i' \in T_{\text{big}}} g_\ell(i)g_\ell(i')Y_{i'}x_{i'} + \sum_{i' \in T_{\text{small}}} g_\ell(i)g_\ell(i')Y_{i'}x_{i'} \]

\[ |Z_\ell - x_i| \geq \frac{\epsilon}{\sqrt{k}}\text{err}_2^k(x) \] implies
- \( A_\ell \) happens (that is some big coordinate collides with \( i \) in \( h_\ell \) or
- \( |Z'_\ell| \geq \frac{\epsilon}{\sqrt{k}}\text{err}_2^k(x) \)

Therefore, by union bound,

\[ \Pr\left[ |Z_\ell - x_i| \geq \frac{\epsilon}{\sqrt{k}}\text{err}_2^k(x) \right] \leq \epsilon^2/3 + 1/3 \leq 2/5 \]

if \( \epsilon \) is sufficiently small.
High probability estimate

**Lemma**

\[
\Pr \left[ |Z_l - x_i| \geq \frac{\epsilon}{\sqrt{k}} \text{err}_2^k(x) \right] \leq 2/5.
\]

Recall \( \tilde{x}_i = \text{median}\{g_1(i)C[1, h_1(i)], \ldots, g_d(i)C[d, h_d(i)]\} \).

- Hence by Chernoff bounds with \( d = \Omega(\log n) \),
  \[
  \Pr \left[ |\tilde{x}_i - x_i| \geq \frac{\epsilon}{\sqrt{k}} \text{err}_2^k(x) \right] \leq 1/n^2
  \]
- By union bound, with probability at least \( (1 - 1/n) \),
  \[
  |\tilde{x}_i - x_i| \leq \frac{\epsilon}{\sqrt{k}} \text{err}_2^k(x) \quad \text{for all } i \in [n].
  \]
High probability estimate

Lemma

\[ \Pr \left[ |Z_l - x_i| \geq \frac{\epsilon}{\sqrt{k}} \text{err}_2^k(x) \right] \leq \frac{2}{5}. \]

Recall \( \tilde{x}_i = \text{median}\{g_1(i)C[1, h_1(i)], \ldots, g_d(i)C[d, h_d(i)]\} \).

- Hence by Chernoff bounds with \( d = \Omega(\log n) \),
  \[ \Pr \left[ |\tilde{x}_i - x_i| \geq \frac{\epsilon}{\sqrt{k}} \text{err}_2^k(x) \right] \leq \frac{1}{n^2} \]
- By union bound, with probability at least \( 1 - 1/n \),
  \[ |\tilde{x}_i - x_i| \leq \frac{\epsilon}{\sqrt{k}} \text{err}_2^k(x) \] for all \( i \in [n] \).

Lemma

Count-Sketch with \( w = \frac{3k}{\epsilon^2} \) and \( d = O(\log n) \) ensures that \( \forall i \in [n], \ |\tilde{x}_i - x_i| \leq \frac{\epsilon}{\sqrt{k}} \text{err}_2^k(x) \) with high probability (at least \( 1 - 1/n \)).
Lemma

Let \(x, y \in \mathbb{R}^n\) such that \(\|x - y\|_\infty \leq \frac{\epsilon}{\sqrt{k}}\text{err}^k_2(x)\). Then, \(\|x - z\|_2 \leq (1 + 5\epsilon)\text{err}^k_2(x)\), where \(z\) is the vector obtained as follows: \(z_i = y_i\) for \(i \in T\) where \(T\) is the set of \(k\) largest (in absolute value) indices of \(y\) and \(z_i = 0\) for \(i \notin T\).

What the lemma is saying:

- \(\tilde{x}\) the estimated vector of Count-Sketch approximates \(x\) very closely in each coordinate
- Algorithm picks the top \(k\) coordinates of \(\tilde{x}\) to create \(z\)
- Then \(z\) approximates \(x\) well
\( x = (100, 100, 100, 100, 95, 0.1, 0.001, 0, 1, 0.5, \ldots) \)

\( k = h \)

\( \bar{x} = (97, 105, 110, 83, 0.5, 0.1, 3, 0, 0 \ldots 2) \)

\[ |\bar{x} - x|_{b_0} \leq \frac{\varepsilon}{\sqrt{k}} = 0 \]

\[ X = \frac{\ell}{\ell} \]

\[ \bar{x} = (97, 83, 99, 105, 90, \ldots) \]

\[ |\bar{x} - x_i| \leq \frac{\varepsilon}{\sqrt{k}} \]

\[ 100^2 \leq t - \]
Second lemma of outline

Lemma

Let $x, y \in \mathbb{R}^n$ such that $\|x - y\|_\infty \leq \frac{\epsilon}{\sqrt{k}} \text{err}^k_2(x)$. Then, $\|x - z\|_2 \leq (1 + 5\epsilon) \text{err}^k_2(x)$, where $z$ is the vector obtained as follows: $z_i = y_i$ for $i \in T$ where $T$ is the set of $k$ largest (in absolute value) indices of $y$ and $z_i = 0$ for $i \notin T$.

What the lemma is saying:

- $\tilde{x}$ the estimated vector of Count-Sketch approximates $x$ very closely in each coordinate
- Algorithm picks the top $k$ coordinates of $\tilde{x}$ to create $z$
- Then $z$ approximates $x$ well

Proof is basically follows the intuition of triangle inequality
Proof of lemma

$S$ (previously $T_{\text{big}}$) is set of $k$ biggest coordinates in $x$
$T$ is the set of $k$ biggest coordinates in $y = \tilde{x}$

Let $E = \frac{1}{\sqrt{k}} \text{err}^k_2(x)$ for ease of notation.

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

Want to bound

$$\|x - z\|_2^2 = \sum_{i \in T} |x_i - z_i|^2 + \sum_{i \in S \setminus T} |x_i - z_i|^2 + \sum_{i \in [n] \setminus (S \cup T)} x_i^2$$

$$= \sum_{i \in T} |x_i - y_i|^2 + \sum_{i \in S \setminus T} x_i^2 + \sum_{i \in [n] \setminus (S \cup T)} x_i^2.$$
Want to bound

$$\|x - z\|^2 = \sum_{i \in T} |x_i - z_i|^2 + \sum_{i \in S \setminus T} |x_i - z_i|^2 + \sum_{i \in [n] \setminus (S \cup T)} x_i^2$$

$$= \sum_{i \in T} |x_i - y_i|^2 + \sum_{i \in S \setminus T} x_i^2 + \sum_{i \in [n] \setminus (S \cup T)} x_i^2.$$ 

First term: $\sum_{i \in T} |x_i - \tilde{x}_i|^2 \leq k \epsilon^2 E^2 \leq \epsilon^2 (\text{err}_k^2(x))^2$
Analysis continued

Want to bound

$$\|x - z\|_2^2 = \sum_{i \in T} |x_i - z_i|^2 + \sum_{i \in S \setminus T} |x_i - z_i|^2 + \sum_{i \in [n] \setminus (S \cup T)} x_i^2$$

$$= \sum_{i \in T} |x_i - y_i|^2 + \sum_{i \in S \setminus T} x_i^2 + \sum_{i \in [n] \setminus (S \cup T)} x_i^2.$$

First term: $\sum_{i \in T} |x_i - \tilde{x}_i|^2 \leq k\epsilon^2 E^2 \leq \epsilon^2 (err_k^2(x))^2$

Third term: common to expression for $(err_k^2(x))^2$
Analysis continued

Want to bound

\[ \| x - z \|_2^2 = \sum_{i \in T} |x_i - z_i|^2 + \sum_{i \in S \setminus T} |x_i - z_i|^2 + \sum_{i \in [n] \setminus (S \cup T)} x_i^2 \]

\[ = \sum_{i \in T} |x_i - y_i|^2 + \sum_{i \in S \setminus T} x_i^2 + \sum_{i \in [n] \setminus (S \cup T)} x_i^2. \]

First term: \[ \sum_{i \in T} |x_i - \tilde{x}_i|^2 \leq k\varepsilon^2 E^2 \leq \varepsilon^2 (\text{err}_2^k(x))^2 \]

Third term: common to expression for \( (\text{err}_2^k(x))^2 \)

Second term: needs more care
Want to bound $\sum_{i \in S \setminus T} x_i^2$

Let $\ell = |S \setminus T| \leq k$. Since $|S| = |T| = k$, $|T \setminus S| = \ell$

Coordinates in $S \setminus T$ and $T \setminus S$ must be close: within $\frac{\epsilon}{\sqrt{k}} \text{err}_2^k(x)$
Analysis contd

Want to bound $\sum_{i \in S \setminus T} x_i^2$

Let $\ell = |S \setminus T| \leq k$. Since $|S| = |T| = k$, $|T \setminus S| = \ell$

Coordinates in $S \setminus T$ and $T \setminus S$ must be close: within $\frac{\epsilon}{\sqrt{k}} err^k_2(x)$

Claim: Let $a = \max_{i \in S \setminus T} |x_i|$ and $b = \min_{i \in T \setminus S} |x_i|$. Then $a \leq b + 2\frac{\epsilon}{\sqrt{k}} err^k_2(x)$. 
Want to bound $\sum_{i \in S \setminus T} x_i^2$

Let $\ell = |S \setminus T| \leq k$. Since $|S| = |T| = k$, $|T \setminus S| = \ell$

Coordinates in $S \setminus T$ and $T \setminus S$ must be close: within $\frac{\epsilon}{\sqrt{k}} \text{err}_2^k(x)$

Claim: Let $a = \max_{i \in S \setminus T} |x_i|$ and $b = \min_{i \in T \setminus S} |x_i|$. Then $a \leq b + 2 \frac{\epsilon}{\sqrt{k}} \text{err}_2^k(x)$.

Therefore

$$\sum_{i \in S \setminus T} x_i^2 \leq \ell a^2 \leq \ell (b + 2 \frac{\epsilon}{\sqrt{k}} \text{err}_2^k(x))^2$$

$$\leq \ell b^2 + 4k \frac{\epsilon^2}{k} (\text{err}_2^k(x))^2 + 4kb \frac{\epsilon}{\sqrt{k}} \text{err}_2^k(x).$$
Analysis contd

\[
\sum_{i \in S \setminus T} x_i^2 \leq \ell a^2 \leq \ell (b + 2 \frac{\epsilon}{\sqrt{k}} \text{err}_k^2(x))^2
\]

\[
\leq \ell b^2 + 4k \frac{\epsilon^2}{k} (\text{err}_2^k(x))^2 + 4kb \frac{\epsilon}{\sqrt{k}} \text{err}_2^k(x)
\]

\[
\leq \ell b^2 + 4\epsilon^2 (\text{err}_2^k(x))^2 + 4\epsilon (\sqrt{k}b) \text{err}_2^k(x)
\]

\[
\leq \ell b^2 + 8\epsilon (\text{err}_2^k(x))^2
\]

\[
\leq \sum_{i \in T \setminus S} x_i^2 + 8\epsilon (\text{err}_2^k(x))^2.
\]

**Exercise:** Why is \( \sqrt{k}b \leq \text{err}_2^k(x) \)? (We used it above.)
\[ \| x - z \|_2^2 = \sum_{i \in T} |x_i - z_i|^2 + \sum_{i \in S \setminus T} |x_i - z_i|^2 + \sum_{i \in [n] \setminus (S \cup T)} x_i^2 \]

\[ = \sum_{i \in T} |x_i - y_i|^2 + \sum_{i \in S \setminus T} x_i^2 + \sum_{i \in [n] \setminus (S \cup T)} x_i^2. \]

First term: \( \sum_{i \in T} |x_i - \tilde{x_i}|^2 \leq k \epsilon^2 E^2 \leq \epsilon^2 (err_k^2(x))^2 \)

Third term: common to expression for \( (err_k^2(x))^2 \)

Second term: at most \( \sum_{i \in T \setminus S} x_i^2 + 8 \epsilon (err_k^2(x))^2 \)

Hence

\[ \| x - z \|_2^2 \leq (1 + 9 \epsilon)(err_k^2(x))^2 \]

Implies

\[ \| x - z \|_2 \leq (\sqrt{1 + 9 \epsilon})err_k^2(x) \leq (1 + 5 \epsilon)err_k^2(x) \]
Application to signal processing

Given signal $x$ approximate it via small number of basis signals

- Fourier analysis and Wavelets
- Useful in compression of various kinds

Given signal $x$ approximate it via small number of basis signals

- Fourier analysis and Wavelets
- Useful in compression of various kinds
Application to signal processing

Given signal $x$ approximate it via small number of basis signals

- Fourier analysis and Wavelets
- Useful in compression of various kinds

Transform $x$ into $y = Bx$ where $B$ is a transform and then approximate $y$ by $k$-sparse vector $z$

To (approximately) reconstruct $x$, output $x' = B^{-1}z$

If $Bx$ can be computed in streaming fashion from stream for $x$, we can apply preceding algorithm to obtain $z$
Compressed Sensing

We saw that *given* \( x \) in streaming fashion we can construct sketch that allows us to find \( k \)-sparse \( z \) that approximates \( x \) with high probability.

**Compressed sensing:** we want to create projection matrix \( \Pi \) such that for *any* \( x \) we can create from \( \Pi x \) a good \( k \)-sparse approximation to \( x \).

Doable! With \( \Pi \) that has \( O(k \log(n/k)) \) rows. Creating \( \Pi \) requires randomization but once found it can be used. Called RIP matrices. First due to Candes, Romberg, Tao and Donoho. Lot of work in signal processing and algorithms.