# Summaries of Streaming Data

Martin J. Strauss
University of Michigan

# Sparse Approximation

National retailer sees a stream of transactions:

- 2 Thomas sold, 1 Thomas returned, 1 TSP sold ...

Implies vector $x$ of item frequencies:

- 40 Thomas, 2 Lego, $-30$ TSP, ...

Goal: Track items with large-magnitude counts

# Example Algorithm

Measurements — Signal, $x$

Measurement matrix, $\Phi$

$$\begin{pmatrix} 5.3 \\ \cdots \\ 0 \\ 5.3 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 0 \\ 5.3 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Recover position and coefficient of single spike in signal.

# Algorithmic Constraints

- Little time per item
- Little storage space
- Little time to answer queries

# Fundamental Queries

Identification: Output a set that

- contains all “heavy” indices
- contains no “light” indices
- (medium weight: no constraint)

Estimation

- estimate large coefficients reliably.

# Summaries

Fundamental queries can be used to build summaries:

- Fourier/Wavelet summaries
- Piecewise-constant, piecewise-linear summaries
- ...

Other user queries can be answered from summary

# Overview of Summaries

- Heavy Hitters
- Weak greedy sparse recovery
- Orthonormal change of basis
- Haar Wavelets
- Histograms (piecewise constant)
- Multi-dimensional (hierarchical)
- Piecewise-linear
- Range queries

# Setup

Design

- a matrix $\Phi$ and decoding algo $D$ that work together.

Process Stream:

- Track $y = \Phi x$.

Answer queries:

- Output $D(\Phi x)$.

# Processing Items

- See “add $v$ to $x_i$”
- Read as “add vector $ve_i$ to $x$”

$$\begin{cases} y \leftarrow \Phi x \\ x \leftarrow x + ve_i \\ y \leftarrow y + v\Phi e_i \end{cases}$$

# Some Costs

Space:

- $|y|$ plus space to store $\Phi$.

Time per item:

- generate $\Phi e_i$
- Usually about proportional to $|y|$
- Sometimes much less if $\Phi$ is sparse

(Still need to analyze time for queries. Depends a lot on $\Phi$ and $D$.)

# Warmup: One Spike, Low Noise

$$\begin{pmatrix} 5.6 \\ \cdots \\ 0.2 \\ 5.5 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 0.1 \\ 0 \\ 5.3 \\ 0 \\ 0 \\ 0 \\ 0.2 \\ 0 \end{pmatrix}$$

$d$ columns and $\log(d)$ rows. (Deterministic and efficient)

If $b^\ell$ is $\ell$'th row of matrix, and spike is at $i$, need

$|x_i| \geq 2.01 \sum j \neq i |x_j|$ or (weaker) $\forall \ell$

$$|x_i| > 2.01 \left| \sum_{j \neq i} b_j^\ell x_j \right| .$$

# Many Spikes? Group Testing

Example:

- 150 soldiers; 3 have syphilis
- Pool specimens into 6 random groups.
- "Many" groups have
  - exactly one sick soldier
  - about 1/6 of the dilution from healthy soldiers
- Perform 6 tests
  - clear $\geq 3$ groups—75 soldiers

# Warmup II: L1 significance

Problem:

- Suppose $|x_i| > \frac{1}{k}\sum_{j \neq i} |x_j|$. Find $i$.

Solution: Hash...

- Keep $\frac{1}{12k}$ fraction of positions at random
  - i.e., consider $xr$, where $r$ is 0/1-valued
- With prob $\geq \frac{1}{12k}$, we keep $i$; i.e., $r_i = 1$.
- For each $j \neq i$, $E[|r_j x_j|] = \frac{1}{k}|x_j|$.

# Warmup II: L1 significance

So

$$E\left[\sum_{j\neq i}|r_jx_j|\right] = \sum_{j\neq i}E[|r_jx_j|] = \frac{1}{12k}\sum_{j\neq i}|x_j|$$

So, with prob $\geq 3/4$ (independently of whether $r_i = 1$)

$$\sum_{j\neq i}|r_jx_j| \leq \frac{1}{3k}\sum_{j\neq i}|x_j| < \frac{1}{3}|x_ir_i|.$$

Repeat, and proceed as above!

# Digression: Linearity of Expectation

Recall that a random variable is a function on a sample space.

$$\begin{aligned} X : \Omega &\to \mathbb{R} \\ \omega &\mapsto X(\omega) \end{aligned}$$

Then $E[X] = \sum_{\omega \in \Omega} X(\omega) \Pr(\omega)$, and so

$$\begin{aligned} E[X+Y] &= \sum_{\omega \in \Omega} (X(\omega) + Y(\omega)) \Pr(\omega) \\ &= \sum_{\omega \in \Omega} X(\omega) \Pr(\omega) + \sum_{\omega \in \Omega} Y(\omega) \Pr(\omega) \\ &= E[X] + E[Y]. \end{aligned}$$

# Digression: Markov

Theorem: If $X$ is a non-negative random variable and $a > 0$, then

$$\Pr(X \geq a) \leq E[X]/a.$$

Proof:

$$\begin{aligned} E[X] &= \sum_x x \Pr(X = x) \\ &\geq \sum_{x \geq a} a \Pr(X = x) \\ &= a \Pr(X \geq a). \end{aligned}$$

E.g., $\Pr(X \geq 4E[X]) \leq 1/4$.

# Repeat

$$\begin{aligned} \Pr(\text{success}) &\geq \frac{3}{4} \cdot \frac{1}{4k} = \frac{3}{16k} > \frac{1}{6k} \\ \Pr(\text{failure}) &< 1 - \frac{1}{6k}. \end{aligned}$$

Repeat $6k$ times, independently.

$$\Pr(\text{all failures}) < \left(1 - \frac{1}{6k}\right)^{6k} \approx 1/e \approx .37 < .5.$$

Repeat total of $6km$ times.

- Modest cost.
- $\Pr(\text{all failures}) < 2^{-m}$.

# Putting it together

Collect repeated $r$'s into matrix, $R$.

Take *row tensor product* $R \otimes_{\mathrm{r}} B$ with bit testing matrix, $B$:

- rows are $\{rb : r \text{ is row of } R, b \text{ is row of } B\}$

# Row Tensor Product, E.g.

$$R = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{pmatrix}, \quad B = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{pmatrix},$$

so

$$B \otimes_{\mathrm{r}} R = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

# Warmup III: L2 significance

Problem: Suppose now that $x_i^2 > \frac{1}{k'} \sum_{j \neq i} x_j^2$; want to find i.

- Note: stronger statement than before.

Solution:

- Multiply each $x_i$ by random $\pm 1$ first
- Keep $\frac{1}{36k'}$, at random
- i.e., consider $rsx$, where
  - $s$ has random signs
  - $r$ is random mask

# Warmup III: L2 significance

Still keep $i$ with prob'y $\frac{1}{12k'}$ (Assume this.)

$$E\left[\left(\sum_{j\neq i} b_j r_j s_j x_j\right)^2\right] = E\left[\sum_{j,\ell\neq i} b_j b_\ell r_j r_\ell s_j s_\ell x_j x_\ell\right]$$

$$= \quad E_r\left[\sum_{j,\ell\neq i} E_s[s_j s_\ell] r_j r_\ell b_j b_\ell x_j x_\ell\right]$$

$$= \quad E_r\left[\sum_{j\neq i, b_j=1} r_j x_j^2\right]$$

$$= \quad \sum_{j\neq i, b_j=1} E[r_j] x_j^2 = \frac{1}{12k'} \sum_{j\neq i, b_j=1} x_j^2 < \frac{1}{12} x_i^2.$$

# Warmup III: L2 significance

With prob $\geq 3/4$,

$$\left(\sum_{j \neq i} b_j r_j s_j x_j\right)^2 < \frac{1}{9} x_i^2,$$

or

$$\sum_{j \neq i} b_j r_j s_j x_j < \frac{1}{3} |r_i s_i x_i|.$$

(Extra repetitions are needed to make **all** $b^\ell$ work simultaneously.)

Proceed as above.

# Digression: Expectation of a product

Theorem: If $X$ and $Y$ are independent, then $E[XY] = E[X]E[Y]$.

Proof:

$$
\begin{aligned}
E[XY] &= \sum_{x,y} xy \Pr(X = x \text{ and } Y = y) \\
&= \sum_{x,y} xy \Pr(X = x) \Pr(Y = y) \\
&= E[X]E[Y].
\end{aligned}
$$

# Digression: Cauchy-Schwarz Inequality

Theorem:

$$\frac{1}{d}\left(\sum_{i=1}^{d}|x_i|\right)^2 \leq \sum_{i=1}^{d} x_i^2 \leq \left(\sum_{i=1}^{d}|x_i|\right)^2 ;$$

either equality is possible.

# Cauchy-Schwarz Inequality: Implication

Thus, if $|x_i| > \sum_{j \neq i} |x_j|$ then

$$|x_i|^2 > \left( \sum_{j \neq i} |x_j| \right)^2 > \sum_{j \neq i} x_i^2.$$

But, if $|x_i|^2 > \sum_{j \neq i} |x_j|^2$, then all we know is

$$|x_i| > \sqrt{\sum_{j \neq i} x_i^2} > \frac{1}{\sqrt{d}} \sum_{j \neq i} |x_j|.$$

Weaker by the large factor $\sqrt{d}$.

# Cauchy-Schwarz Inequality: Proof

For $\sum x_i^2 \leq (\sum |x_i|)^2$:

$$\sum_i x_i^2 \leq \sum_{i,j} x_i x_j = \left( \sum_i x_i \right)^2 .$$

Pick out diagonal; Equality if there is only one term.

# Cauchy-Schwarz Inequality: Proof

For $\frac{1}{d}\left(\sum |x_i|\right)^2 \le \sum x_i^2$, need

$$\sum_{i=1}^{d} x_i = \langle x, 1 \rangle \le \|x\| \cdot \|1\| = \|x\| \cdot \sqrt{d}.$$

We'll show $\langle x, y \rangle \le \|x\| \, \|y\|$.

Can normalize; assume $\|x\| = \|y\| = 1$. Then

$$0 \le \langle x - y, x - y \rangle = \|x\|^2 + \|y\|^2 - 2 \langle x, y \rangle .$$

So $\langle x, y \rangle \le \left(\|x\|^2 + \|y\|^2\right)/2 = 1 = \|x\| \cdot \|y\|$. Equality if (and only if) $x$ and $y$ are proportional.

# On to Estimation

Let $s$ be a random $\pm 1$-valued random vector.

Atomic estimator for $x_i$ is $X = s_i \langle x, s \rangle$. Then

$$X = s_i \sum_j s_j x_j = \sum_j s_i s_j x_j,$$

so

$$E[X] = \sum_j E[s_i s_j] x_j = x_i.$$

Need to bound variance.

# Estimation: Variance

Also

$$\begin{aligned}
\operatorname{var}(X) &= E[X^2] - x_i^2 \\
&= E\left[\sum_{j,\ell} s_j s_\ell x_j x_\ell\right] \\
&= \sum_{j,\ell} E\left[s_j s_\ell\right] x_j x_\ell \\
&= \sum_{j \neq i} x_j^2 .
\end{aligned}$$

Standard deviation small/bounded in terms of target value.

# Markov/Chebychev

Theorem: For $a > 0$,

$$\Pr(|X - E[X]| \geq a) \leq \mathrm{var}(X)/a^2.$$

Proof:

$$\Pr((X - E[X])^2 \geq a^2) \leq \mathrm{var}(X)/a^2.$$

Get $\Pr(|X - x_i| \geq 3||x||) \leq 1/9$.

# Better distortion

Let $Y$ be the average of m copies of $X$. Then $E[Y] = E[X]$ and $\text{var}(Y) = \frac{1}{m}\text{var}(X)$.

Get

$$\Pr\left(|Y - x_i| \geq \frac{3}{m}\|x\|\right) \leq \frac{1}{9}.$$

# Digression: Improving Variance

Theorem: Let $Y$ be the average of $m$ copies of $X$. Then $\mathrm{var}(Y) = \frac{1}{m}\mathrm{var}(X)$. Proof:

Let $\mu = E[X] = E[Y]$.

Then $E[X - \mu] = 0$ and

$$\mathrm{var}(X - \mu) = E[(X - \mu - 0)^2] = \mathrm{var}(X).$$

# Digression: Improving Variance

So assume $E[X] = E[Y] = 0$. Then

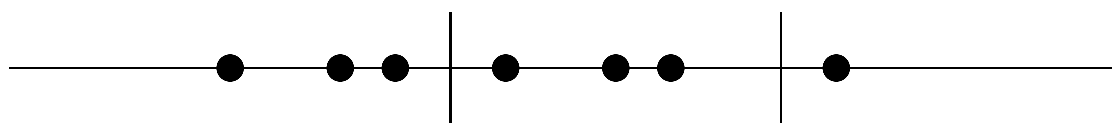
$$\begin{aligned} \mathrm{var}(Y) = E[Y^2] &= E\left[\left(\frac{1}{m}\sum X_i\right)^2\right] \\ &= \frac{1}{m^2}\sum_{i,j} E[X_i X_j], \quad \text{using independence} \\ &= \frac{1}{m^2}\sum_i E[X_i^2] \\ &= \frac{1}{m}E[X^2]. \end{aligned}$$

# Better failure probability.

Theorem: Suppose $\Pr(Y \text{ is bad}) < 1/9$.

Let $Z$ be the median of $l$ independent copies of $Y$. Then $\Pr(Z \text{ is bad}) < 2^{-\Omega(l)}$.

Proof: $Z$ is bad only if at least half of the $Y$'s are bad. Apply Chernoff.



# Digression: Chernoff Bounds

Theorem: Suppose each of $n$ $Y_i$'s is independent with

$$Y_i = \begin{cases} 1-p, & \text{with probability } p; \\ -p, & \text{with probability } 1-p. \end{cases}$$

Let $Y = \sum_i Y_i$. If $a > 0$, then

$$\Pr(Y > a) < e^{-2a^2/n}.$$

# Chernoff: Proof

(Just for $p = 1/2$, so $Y_i$ is $\pm 1/2$, uniformly.)

Lemma: For $\lambda > 0$, $\frac{e^{\lambda}+e^{-\lambda}}{2} < e^{\lambda^2/2}$. (Proof: Taylor.)

$$
\begin{aligned}
E[e^{2\lambda \sum Y_i}] &= \prod E[e^{2\lambda Y_i}] \\
&= \left(\frac{e^{\lambda} + e^{-\lambda}}{2}\right)^n \\
&< e^{\lambda^2 n/2}.
\end{aligned}
$$

# Chernoff, cont'd

$$\begin{aligned}\Pr(Y > a) &= \Pr\left(e^{2\lambda Y} > e^{2\lambda a}\right)\\ &\leq \frac{E[e^{2\lambda Y}]}{e^{2\lambda a}}\\ &\leq e^{\lambda^2 n/2 - 2\lambda a}.\end{aligned}$$

Put $\lambda = 2a/n$; get

$$\Pr(Y > a) < e^{-2a^2/n}.$$

# To this point

Find all $i$ such that $x_i^2 > \frac{1}{k}\sum_{j \neq i} x_j^2$, with failure probability $2^{-\ell}$.

- Need poly$(k, \ell)$ rows in the matrix $B \otimes_{\mathrm{r}} S \otimes_{\mathrm{r}} R$; comparable runtimes.

Estimate each $x_i$ up to $\pm\epsilon \|x\|$ with failure probability $2^{-\ell}$.

- Need poly$(\ell/\epsilon)$ rows; comparable runtimes.

# Space

To this point, fully random matrices.

- Expensive to store!

But...

- Need only pairwise independence within each row
- (sometimes need full independence from row to row, but this is usually ok).
- i.e., two entries $r_j$ and $r_\ell$ in the same row need to be independent, but three entries may be dependent.
- This can cut down on needed space.

# Pairwise Independence: Construction

Random vector $s$ in $\pm 1^d$ (equivalently, $\mathbb{Z}_2^d$)

Index $i$ is a 0/1 vector of length $\log(d)$, i.e., $i \in \mathbb{Z}_2^{\log(d)}$.

Pick vector $q \in \mathbb{Z}_2^{\log(d)}$ and bit $c \in \mathbb{Z}_2$.

Define $\boxed{s_i = c + \langle q, i \rangle}$ (mod 2).

Then, if $i \neq j$, then $(s_i, s_j)$ takes all four possibilities with equal probability.

# Pairwise Independence: Proof

$s_i$ is uniform because $c$ is random.

Conditioned on $s_i$, $s_j$ is uniform:

- Sufficient to show that $s_i + s_j$ is uniform.
- $s_i + s_j = (c + \langle q, i \rangle) + (c + \langle q, j \rangle) = \langle q, i + j \rangle$
- $i \neq j$, so they differ on some bit, the $\ell$'th.
- As $q_\ell$ varies, $s_i + s_j$ varies uniformly over $\mathbb{Z}_2$.

# Pairwise independence, for $r$

Hashing into one of $k$ buckets. Take $\log(k)$ independent hashes into two buckets. Get bucket label bit-by-bit.

# Space, again

For each row $s$, need only store $q$ and $c$: $\log(d) + 1$ bits.

For each row $r$, need only $\log(k)$ copies of $q$ and $c$: $O(\log(d)\log(k))$ bits.

(Many other constructions are possible.)

# All Together—Heavy Hitters

- Find all $i$ such that $x_i^2 > (1/k)\sum_{j \neq i} x_j^2$, with failure probability $2^{-\ell}$.
- Estimate each $x_i$ up to $\pm\epsilon \|x\|$ with failure probability $2^{-\ell}$.
- Space, time per item, and query time are $\mathrm{poly}(k, \ell, \log(d), 1/\epsilon)$.

# Sparse Recovery

Next topic: Sparse Recovery.

Fix $k$ and $\epsilon$.

Want $\widetilde{x}$ such that

$$\|\widetilde{x} - x\|_2 \leq (1+\epsilon) \left\|x_{(k)} - x\right\|_2 .$$

Here $x_{(k)}$ is best $k$-term approximation to $x$.

Will build on heavy hitters.

# Sparse Recovery: Issue

Suppose $k = 10$ and coefficient magnitudes are $1, 1/2, 1/4, 18, 1/16, \ldots$

Want to find top $k$ terms in time poly$(k)$, not time $2^k$.

Heavy Hitters algorithm only guarantees that we find and estimate well terms with magnitude around $1/k$—about $\log(k)$ terms.

# Weak Greedy Algorithm

- Find indices of heavy terms in $x$
- Estimate their coefs, getting intermediate rep'n $r$.
  - iterative subroutine here
- Recurse on $x - r$.

# Weak Greedy Algorithm

After removing top few terms, others become relatively larger.

Can get sketch $\Phi(x - r)$ as $\Phi x - \Phi r$

At this point, $\widetilde{x}$ may have more than $k$ terms (to be fixed).

*Weak* greedy–may not find the heaviest term.

# Iterative Estimation

Have: a set $I$ of $k$ indices, parameter $\epsilon$

Want: coefficient estimates so that the resulting approximation $\widetilde{x}$ satisfies

$$\|\widetilde{x} - x\| \leq (1+\epsilon)\, \|x - x_I\|\,.$$

Define

- $I^{\mathrm{c}}$ be the complement of $I$.
- $E_I = \sum_{i \in I} |x_i|^2$ be *original* energy in $I$
- $\widetilde{E}_I = \sum_{i \in I} |x_i - \widetilde{x}_i|^2$ to be energy in $I$ *after* one round of estimation.
- $\Delta = E_I / E_{I^{\mathrm{c}}}$ to be the dynamic range.

# Iterative Estimation: Algorithm

Have: a set $I$ of $k$ indices, parameter $\epsilon$

Want: coefficient estimates so that the resulting approximation $\widetilde{x}$ satisfies

$$\|\widetilde{x} - x\| \leq (1 + \epsilon) \|x - x_I\| .$$

Repeat $\log(\Delta/\epsilon)$ times

1. estimate each $x_i$ for $i \in I$, by $\widetilde{x}_i$ with $|\widetilde{x}_i - x_i|^2 < \frac{\epsilon}{2k(1+\epsilon)} \widetilde{E}_i^{\mathrm{c}}$.
2. update $x$.

# Iterative Estimation: Proof

Get: $\widetilde{E}_I \leq \frac{\epsilon}{2(1+\epsilon)}(E_I + E_{I^c})$.

Case $E_I > \epsilon \cdot E_{I^c}$:

$$\begin{aligned}\widetilde{E}_I &\leq \frac{\epsilon}{2(1+\epsilon)}\left(E_I + E_{I^c}\right)\\ &\leq \frac{\epsilon}{2(1+\epsilon)}E_I + \frac{1}{2(1+\epsilon)}E_I\\ &= \frac{1}{2}E_I.\end{aligned}$$

Geometric improvement. Get down to $\epsilon E_{I^c}$ if this case holds for all iterations.

# Iterative Estimation: Proof

Case $E_I \leq \epsilon \cdot E_{I^c}$:

$$\begin{aligned} \widetilde{E}_I &\leq \frac{\epsilon}{2(1+\epsilon)} \left(E_I + E_{I^c}\right) \\ &\leq \frac{\epsilon}{2} E_{I^c}. \end{aligned}$$

$E_I$ fluctuates only in the range 0 to $\frac{\epsilon}{2} E_{I^c}$ after dropping below $\epsilon E_{I^c}$.

# Iterative Identification

Similar to estimation

Repeat $\log(\Delta/\epsilon)$ times

1. Identify indices $i$ with $|x_i|^2 > \frac{\epsilon}{4k(1+\epsilon)}\widetilde{E}_{i^c}$.
2. Estimate each $x_i$, for $i \in I$, by $\widetilde{x}_i$ with $\widetilde{E}_I \le E_{I^c}$
3. update $x$.

Final estimation:

- $\widetilde{E}_I \le \frac{\epsilon}{3} E_{I^c}$.

# Iterative Identification: Proof

First: Estimation errors do not substantially affect Identification.

Issue:

- Have a set $I$ of indices for intermediate $r$.
- We'll identify positions in $x - r$.
- Values in $(x - r)_I$ are based on estimates and may be larger than $x_I$
- ...contribute extra noise; obstacle to identification.

Identify $i$ if $|x_i|^2$ large compared with $\widetilde{E}_{i^c}$, so get $i$ if $|x_i|^2$ large compared with

$$E_I > (1 - \epsilon)\widetilde{E} > (1 - \epsilon)\widetilde{E}_{i^c}.$$

# Iterative Identification: Proof

Among top k, miss a total of at most

$$E_{K\setminus I} \leq \frac{\epsilon}{2(1+\epsilon)} E = \frac{\epsilon}{2(1+\epsilon)} (E_K + E_{K^c}).$$

Case $E_K > \epsilon E_{K^c}$:

$$\begin{aligned} E_{K\setminus I} &\leq \frac{\epsilon}{2(1+\epsilon)} (E_K + E_{K^c}) \\ &< \frac{\epsilon}{2(1+\epsilon)} E_K + \frac{1}{2(1+\epsilon)} E_K \\ &= \frac{1}{2} E_K. \end{aligned}$$

# Iterative Identification: Proof

Case $E_K \le \epsilon E_{K^c}$:

$$\begin{aligned} E_{K \setminus I} &\le \frac{\epsilon}{2(1+\epsilon)}(E_K + E_{K^c}) \\ &\le \frac{\epsilon}{2} E_{K^c}. \end{aligned}$$

Either case, identify enough.

# Iterative Identification—proof

Three sources of error:

1. outside top $k$—excusable.
2. inside top $k$, but not found—small compared with excusable.
3. found, and estimated incorrectly—small compared with excusable.

# Exactly $k$ Terms Output

Algorithm:

1. Get $\widetilde{x}$ with $\|\widetilde{x} - x\|^2 \le (1+\epsilon)\left\|x_{(k)} - x\right\|^2$.
2. Estimate each $x_i$ by $\widetilde{x}_i$ with $|x_i - \widetilde{x}_i|^2 \le \frac{\epsilon^2}{k} E_{K^c}$.
3. Output top $k$ terms of $\widetilde{x}$, i.e., $\widetilde{x}_{(k)}$

# Exactly $k$ Terms Output: Proof

Sources of error:

1. Terms in $K \setminus I$ (small; already shown)
2. Error in terms we do take (small; already shown)
3. Error from mis-ranking
   - if $k + 1$ terms are about equally good, we won't know for sure which are the $k$ biggest.

# Exactly $k$ Terms Output: Misranking

Idea: only displace one term for another if their magnitudes are close. Some care needed to keep quadratic dependence on $\epsilon$.

Let $y$ be a vector of terms in top $k$ that are displaced by an equal number of terms *not* in the top $k$, the vector $z$. Both $y$ and $z$ have length at most $k$. $y_i$ is displaced by $z_i$.

Assume we have found and estimated all terms in $y$ (else don't care; these terms are small.)

# Exactly $k$ Terms Output: Proof

By the triangle inequality,

$$\begin{aligned} |y_i| &\le |\widetilde{y}_i| + |y_i - \widetilde{y}_i| \\ |z_i| &\ge |\widetilde{z}_i| - |z_i - \widetilde{z}_i| \end{aligned}$$

Thus

$$\begin{aligned} |y_i| - |z_i| &\le |\widetilde{y}_i| - |\widetilde{z}_i| + |y_i - \widetilde{y}_i| + |z_i - \widetilde{z}_i| \\ &\le |y_i - \widetilde{y}_i| + |z_i - \widetilde{z}_i| \\ &\le 2\epsilon\sqrt{E_{K^c}/k} \end{aligned}$$

Thus

$$\| |y| - |z| \| \le 2\epsilon\sqrt{E_{K^c}}.$$

# Exactly $k$ Terms Output: Proof

Continuing...

$$\begin{aligned} \| |z| \| &= \|z\| \leq \sqrt{E_{K^c}} \\ \| |y| \| &= \|y\| \leq \|z\| + \| |y| - |z| \| , \end{aligned}$$

so

$$\begin{aligned} \| |y| + |z| \| &\leq 2\|z\| + \| |y| - |z| \| \\ &\leq 2\sqrt{E_{K^c}} + 2\epsilon\sqrt{E_{K^c}} \\ &\leq 3\sqrt{E_{K^c}}, \end{aligned}$$

# Exactly $k$ Terms Output: Proof

so, finally,

$$\begin{aligned} \|y\|^2 - \|z\|^2 &= \||y|\|^2 - \||z|\|^2 \\ &= \langle |y| + |z|, |y| - |z| \rangle \\ &\le \||y| + |z|\| \cdot \||y| - |z|\| \\ &\le 3\sqrt{E_{K^c}} \cdot 2\epsilon\sqrt{E_{K^c}} \\ &\le 6\epsilon E_{K^c}. \end{aligned}$$

# Overview of Summaries

- Heavy Hitters
- Weak greedy sparse recovery
- Orthonormal change of basis
- Haar Wavelets
- Histograms (piecewise constant)
- Multi-dimensional (hierarchical)
- Piecewise-linear
- Range queries

# Finding Other Heavy Things

E.g., Fourier coefficients.

Important by themselves

Useful toward other kinds of summaries

# Orthonormal bases

Columns of $U$ is ONB if columns of $U$ are perpendicular and unit Euclidean length. Thus

$$\langle \psi_j, \psi_k \rangle = \begin{cases} 1, & j = k \\ 0, & \text{otherwise.} \end{cases}$$

E.g.:

- Fourier basis
- Haar wavelet basis

# Decompositions and Parseval

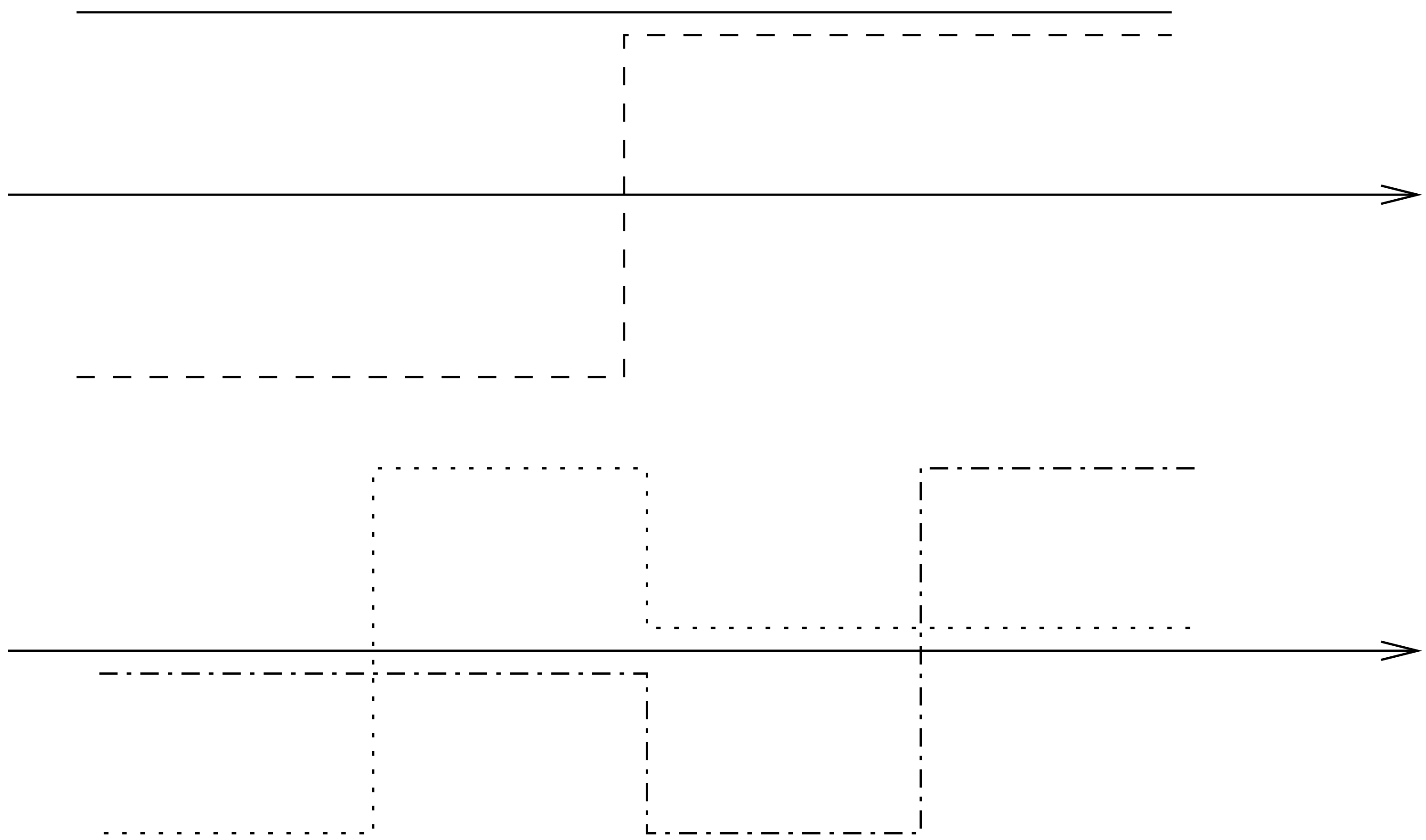
Let $\{\psi_j\}$ be ONB. Then, for any $x$,

$$x = \sum \langle x, \psi_j \rangle \, \psi_j .$$

and

$$\sum_j \langle x, \psi_j \rangle^2 = \sum_i x_i^2$$

# Haar Wavelets, Graphically



E.g.,

$$
\begin{array}{cccccccc}
+1 & +1 & +1 & +1 & +1 & +1 & +1 & +1 \\
-1 & -1 & -1 & -1 & +1 & +1 & +1 & +1 \\
-1 & -1 & +1 & +1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & -1 & +1 & +1 \\
-1 & +1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & +1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & +1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -1 & +1
\end{array}
$$

# Heavy Hitters under Orthonormal Change of Basis

Have vector $x = U\widehat{x}$, where $\widehat{x}$ is sparse

Process stream by transforming $\Phi$:

- Collect $\Phi\widehat{x} = \Phi(U^{-1}U)\widehat{x} = (\Phi U^{-1})\widehat{x}$.

Answer queries:

- Recover heavy hitters in $\widehat{x}$
- Implicitly recover heavy $U$-coefficients of $x$.

Alternatively, transform updates...

# Haar Wavelets—per-Item Time

See “add $v$ to $x_i$”

Want to simulate changes to $\widehat{x} = U^{-1}x$

Regard as “add $v$ to $x_i$” as “add $ve_i$ to $x$”

Decompose $ve_i$ into its Haar wavelet components,

$$ve_i = \sum_j v \langle e_i, \psi_j \rangle \psi_j .$$

Key: $\langle e_i, \psi_j \rangle = 0$ unless $i \in \text{supp}(\psi_j)$.

- Just $O(\log(d))$ such $j$’s—$O(\log(d))$ $\widehat{x}_j$’s change.

# Overview of Summaries

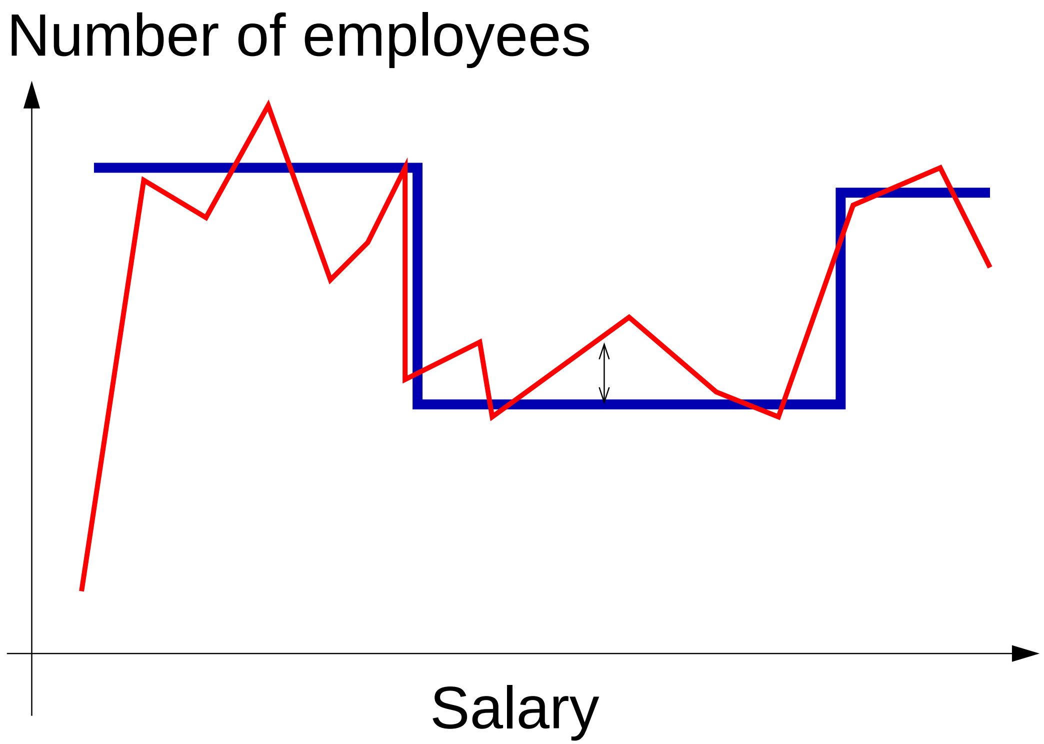
- Heavy Hitters
- Weak greedy sparse recovery
- Orthonormal change of basis
- Haar Wavelets
- Histograms (piecewise constant)
- Multi-dimensional (hierarchical)
- Piecewise-linear
- Range queries

# Histograms

Still see stream of additive updates: "add $v$ to $x_i$" Want $B$-piece piecewise-constant representation, $h$, with

$$\|h - x\| \leq (1+\epsilon)\,\|h_{\text{opt}} - x\|\,.$$

We optimize boundary positions and heights.


Number of employees
Salary

# Histograms–Algorithm Overview

Key idea: Haar wavelets and histograms simulate each other efficiently.

- $t$-term wavelet is $O(t)$-bucket histogram
- $B$-bucket histogram is $O(B\log(d))$-term wavelet rep'n

Next, class of algorithms with varying costs and guarantees:

- Get good Haar representation
- Modify it into a histogram

# Simulation

Histograms simulate Haar wavelets:

- Each Haar wavelet is piecewise constant with 4 pieces (3 breaks), so $t$ terms have $3t$ breaks $(3t+1)$ pieces.

Haar wavelets simulate histograms:

- If $h$ is a $B$-bucket histogram and $\psi_j$'s are wavelets, then
  - ◇ $h = \sum_j \langle h, \psi_j \rangle \psi_j$.
  - ◇ $\langle h, \psi_j \rangle = 0$ unless $\text{supp}(\psi_j)$ intersects a boundary of $h$.
  - ◇ $\leq O(\log(d))$ such wavelets; $\leq O(\log(d))$ terms in a $B$-bucket histogram.

# Algorithm 1

1. Get $O(B \log(d))$-term wavelet rep'n $w$ with

$$\|w - x\| \leq (1+\epsilon) \|h_{\text{opt}} - x\| .$$

2. Return $w$ as a $O(B \log(d))$-bucket histogram

Compared with optimal, $O(\log(d))$ times more buckets and $(1+\epsilon)$ times more error—a $(O(\log(d)), 1+\epsilon)$-approximation.

We can do better...

# Algorithm 2

1. Get $O(B\log(d))$-term wavelet rep'n $w$ with

$$\|w - x\| \le (1+\epsilon)\,\|h_{\text{opt}} - x\|\,.$$

2. Returnn best $B$-bucket histogram $h$ to $w$. (How? soon.)

Get a $(1, 3 + o(1))$-approximation:

$$\begin{aligned}
\|h - x\| &\le \|h - w\| + \|w - x\| \\
&\le \|h_{\text{opt}} - w\| + \|w - x\| \\
&\le \|h_{\text{opt}} - x\| + 2\,\|w - x\| \\
&\le (3 + 2\epsilon)\,\|h_{\text{opt}} - x\|\,,
\end{aligned}$$

# Algorithm 3

1. Get $O(B \log(d) \log(1/\epsilon)/\epsilon^2)$-term wavelet rep'n $w$ with
$$\|w - x\| \leq (1+\epsilon)\,\|h_{\text{opt}} - x\|\,.$$
2. Possibly discard some terms, getting a *robust* $w_{\text{rob}}$.
3. Output best $B$-bucket histogram $h$ to $w_{\text{rob}}$.

Get a $(1, 1+\epsilon)$-approximation. Next:

- What is "robust?"
- Proof of correctness.
- How to find $h$ from $w_{\text{rob}}$.

# Robust Representations

Assume exact estimation (We've shown estimation error is dominated by other error.)

Have $O(B \log(d) \log(1/\epsilon)/\epsilon^2)$-term repn, $w$.

Let $B' = 3B \log(d)$ (hist to wavelet simulation expression)

Consider $w_{(B')}, w_{(2B')}, \ldots$

Let $w_{\text{rob}}$ be

$$w_{\text{rob}} = \begin{cases} w_{(jB')}, & \left\| w_{(jB'..(j+1)B')} \right\|^2 \leq \epsilon^2 \left\| w_{((j+1)B'..)} \right\|^2 \\ w, & \text{otherwise.} \end{cases}$$

"Take terms from top until there is little progress."

# Robust Representation, Continued Progress

Continued progress on $w$ implies very close to $x$.

$\left\| w_{(jB'..(j+1)B')} \right\|^2$ drops exponentially in $j$:

1. Group terms, $2/\epsilon^2$ per group.
2. Each group has twice the energy of the remaining terms, i.e., twice the energy of the remaining groups, so at least twice the energy of the next group.

# Robust Representation, Continued Progress

Terms drop off exponentially. Thus

$$\begin{aligned} \|x - w_{\mathrm{rob}}\|^2 &= \|x - w\|^2 \\ &\le d \left\|w_{(\mathrm{last})}\right\|^2 \\ &\le \epsilon^2 \left\|w_{(B'..2B')}\right\|^2 \\ &\le \epsilon^2 \left\|x - w_{(1..B')}\right\|^2 \\ &\le \epsilon^2 (1+\epsilon) \left\|x - h_{\mathrm{opt}}\right\|^2 \end{aligned}$$

Need $T = (1/\epsilon)^2 \log(d/\epsilon^2)$ repetitions, so

$$(1 - \epsilon^2)^T = \epsilon^2 / d.$$

# Robust Representation, Continued Progress

Note: $\left\|x - w_{(B')}\right\| \le (1+\epsilon)\,\|x - h_{\mathrm{opt}}\|$, i.e., $w_{(B')}$ is accurate enough. (It has too many terms.)

Final guarantee:

$$\begin{aligned}\|h - x\| &\le \|h - w_{\mathrm{rob}}\| + \|w_{\mathrm{rob}} - x\| \\ &\le \|h_{\mathrm{opt}} - w_{\mathrm{rob}}\| + \|w_{\mathrm{rob}} - x\| \\ &\le \|h_{\mathrm{opt}} - x\| + 2\,\|w_{\mathrm{rob}} - x\| \\ &\le (1+3\epsilon)\,\|h_{\mathrm{opt}} - x\|\,.\end{aligned}$$

Adjust $\epsilon$, and we're done.

# Robust Representation, No Progress

No progress on $w$ implies no progress on $x$:

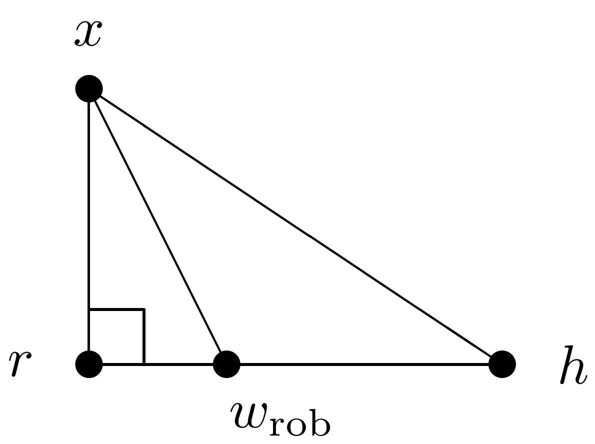
$$\left\|w_{(jB'..(j+1)B')}\right\|^2 \le \epsilon^2 \left\|w_{((j+1)B'..)}\right\|^2$$

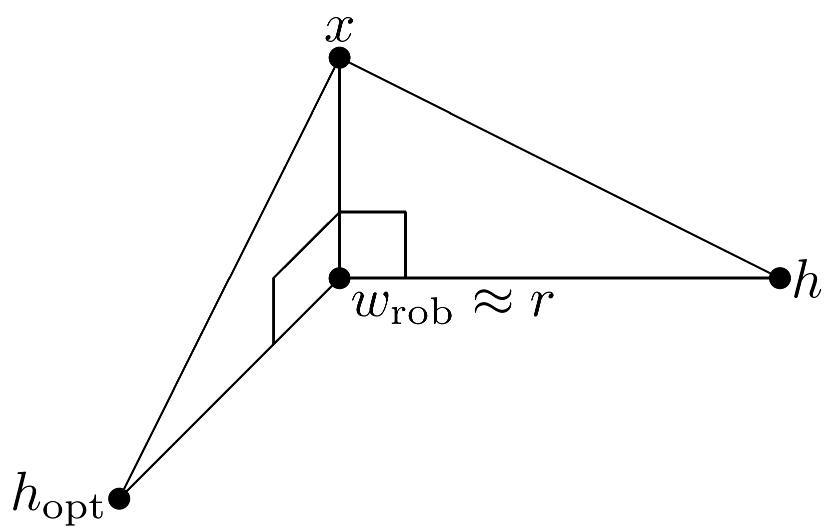
implies

$$\begin{aligned}\left\|w_{(jB'..(j+1)B')}\right\|^2 &\le \epsilon^2 \left\|x_{((j+1)B'..)}\right\|^2 \\ &\le \epsilon^2 \left\|x - h_{\text{opt}}\right\|^2 .\end{aligned}$$

So, the best linear combination, $r$, of $w_{\text{rob}}$ and any $B$-bucket histogram isn't much better than $w_{\text{rob}}$.

# Robust Representation, No Progress




Approximately: $\|h - r\| \leq \|h_{\text{opt}} - r\|$, so $\|h - x\| \leq \|h_{\text{opt}} - x\|$.

# Robust Representation, No Progress

$\|x - w_{\mathrm{rob}}\|$ and $\|w_{\mathrm{rob}} - h_{\mathrm{opt}}\|$ are bounded.

$$\begin{aligned} \|x - w_{\mathrm{rob}}\| &\leq (1+\epsilon)\,\|x - h_{\mathrm{opt}}\| \\ \|w_{\mathrm{rob}} - h_{\mathrm{opt}}\| &\leq (3+\epsilon)3\,\|x - h\|\,. \end{aligned}$$

Also,

$$\|r - w_{\mathrm{rob}}\| \leq \epsilon\,\|x - h_{\mathrm{opt}}\|\,.$$

# Robust Representation, No Progress

We have

$$\begin{aligned}
\|h - x\|^2 &= \|h - r\|^2 + \|r - x\|^2 \\
&\leq (\|h - w_{\text{rob}}\| + \|w_{\text{rob}} - r\|)^2 \\
&\quad +(\|x - w_{\text{rob}}\| - \|w_{\text{rob}} - r\|)^2 \\
&\leq \|h - w_{\text{rob}}\|^2 + \|w_{\text{rob}} - r\|^2 + \|x - w_{\text{rob}}\|^2 \\
&\quad + \|w_{\text{rob}} - r\|^2 + 2\,\|h - w_{\text{rob}}\| \cdot \|w_{\text{rob}} - r\| \\
&\leq \|h_{\text{opt}} - w_{\text{rob}}\|^2 + \|w_{\text{rob}} - r\|^2 + \|x - w_{\text{rob}}\|^2 \\
&\quad + \|w_{\text{rob}} - r\|^2 + 2\,\|h_{\text{opt}} - w_{\text{rob}}\| \cdot \|w_{\text{rob}} - r\| \\
&\leq \|h_{\text{opt}} - w_{\text{rob}}\|^2 + \|x - w_{\text{rob}}\|^2 \\
&\quad +9 \cdot \epsilon \cdot \|x - h_{\text{opt}}\|^2 ,
\end{aligned}$$

# Robust Representation, No Progress

...and, similarly,

$$\begin{aligned}
\|h_{\text{opt}} - x\|^2 &= \|h_{\text{opt}} - r'\|^2 + \|r' - x\|^2 \\
&\geq (\|h_{\text{opt}} - w_{\text{rob}}\| - \|w_{\text{rob}} - r'\|)^2 \\
&\quad +(\|x - w_{\text{rob}}\| - \|w_{\text{rob}} - r'\|)^2 \\
&\geq \|h_{\text{opt}} - w_{\text{rob}}\|^2 + 2\,\|w_{\text{rob}} - r'\|^2 + \|x - w_{\text{rob}}\|^2 \\
&\quad -2\,\|h_{\text{opt}} - w_{\text{rob}}\| \cdot \|w_{\text{rob}} - r'\| \\
&\quad -2\,\|x - w_{\text{rob}}\| \cdot \|w_{\text{rob}} - r'\| \\
&\geq \|h_{\text{opt}} - w_{\text{rob}}\|^2 + \|x - w_{\text{rob}}\|^2 \\
&\quad -9 \cdot \epsilon \cdot \|x - h_{\text{opt}}\|^2 .
\end{aligned}$$

# Robust Representation, No Progress

So

$$\|h - x\|^2 - \|h_{\mathrm{opt}} - x\|^2 \leq 18 \cdot \epsilon \cdot \|x - h_{\mathrm{opt}}\|^2 ,$$

or

$$\|h - x\|^2 \leq (1 + 18\epsilon)\, \|h_{\mathrm{opt}} - x\|^2 .$$

# Warmup: Best Histogram, Full Space

Want best $B$-bucket histogram to $x$. Use dynamic programming, based on the following recursion. Define

- $\mathrm{Err}[j,k]$ = error of best $k$-bucket histogram to $x$ on $[0,j)$.
- $\mathrm{Cost}[j,j']$ = error of best 1-bucket histogram to $x$ on $[j,j')$.

So:

$$\mathrm{Err}[j,k] = \min_{\ell<j} \mathrm{Err}[\ell,k-1] + \mathrm{Cost}[l,j).$$

"$k-1$ buckets on $[0,\ell)$ and one bucket on $[\ell,j)$. Take best $\ell$."

Runtime: $j < d, k < B, l < d$; total $O(d^2B)$.

Can construct actual histogram (not just error) as we go (keep the $\ell$'s that witness the minimization).

# Prefix array

From $x$, construct $Px$: $x_0, x_0 + x_1, x_0 + x_1 + x_2, \ldots$

Also $Px^2$.

Can get $\mathrm{Cost}[\ell, j]$ from $\ell$ and $j$ in constant time:

- $x_\ell + x_{\ell+1} + \cdots + x_{j-1} = (Px)_j - (Px)_\ell$.
- Best height is average $\mu = \frac{1}{j-\ell}\left((Px)_\ell - (Px)_j\right)$.
- Error is $\sum_{\ell \le i < j} (x_i - \mu)^2 = \sum x_i^2 - 2\mu \sum x_i + \mu^2$.

# Best Histogram to Robust Representation

Want best $B$-bucket histogram $h$ to $w_{\mathrm{rob}}$.

wlog, boundaries of $h$ are among boundaries of $w_{\mathrm{rob}}$.

Dynamic programming takes time $O(|w_{\mathrm{rob}}|^2 \cdot B)$, where $|w_{\mathrm{rob}}|$ is the number of boundaries in $w_{\mathrm{rob}}$.

# Overview of Summaries

- Heavy Hitters
- Weak greedy sparse recovery
- Orthonormal change of basis
- Haar Wavelets
- Histograms (piecewise constant)
- Multi-dimensional (hierarchical)
- Piecewise-linear
- Range queries

# Two-Dimensional Histograms

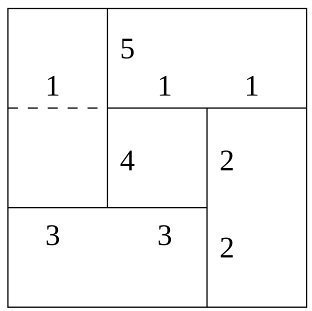
Approximation is constant on rectangles

Hierarchical (recursively split an existing rectangle) or general.

Theorem: Any $B$-bucket (general) partition can be refined into a $(4B)$-bucket hierarchical partition.

Proof omitted; not needed for algorithm.

Aim: $(1, 1+\epsilon)$-approximate hierarchical histogram, which is a $(4, 1+\epsilon)$-approx general histogram.



# 2-D Histograms–Overall Strategy

Same overall strategy as 1-D:

- Find best $B'$-term rep'n over "tensor-product of Haar wavelets."
- Cull back to a robust representation, $w_{\mathrm{rob}}$
- Output best hierarchical histogram to $w_{\mathrm{rob}}$.

Next:

- What is tensor-product of Haar wavelets?
- How to find best B-bucket hierarchical histogram.

# Tensor products

Need ONB that simulates and is simulated by 1-bucket histograms.

Generally: $(\alpha \otimes \beta)(x, y) = \alpha(x)\beta(y)$.

Use tensor product of Haar wavelets:

$$\psi_{j,k}(x, y) = \psi_j(x) \cdot \psi_k(y).$$

Tensor product of ONBs is ONB.

# Processing Updates

Update to $x$ leads to updates to $O(\log^2(d))$ tensor product of Haar wavelets.

(Algorithm is exponential in the dimension, 2.)

# Dynamic Programming

Want best hierarchical $h$ to $w_{\text{rob}}$.

Boundaries of $h$ can be taken from boundaries of $w_{\text{rob}}$.

Best $j$-cut hierarchical $h$ has:

- a full cut (horiz or vert, say vert)
- a $k$-cut partition on the left
- a $(j-1-k)$-cut partition on the right.

Runtime: polynomial in boundaries of $w_{\text{rob}}$ and desired number of buckets.

# Overview of Summaries

- Heavy Hitters
- Weak greedy sparse recovery
- Orthonormal change of basis
- Haar Wavelets
- Histograms (piecewise constant)
- Multi-dimensional (hierarchical)
- Piecewise-linear
- Range queries

# Piecewise-linear representations

Want best $B$-bucket pw-linear approx to $x$.

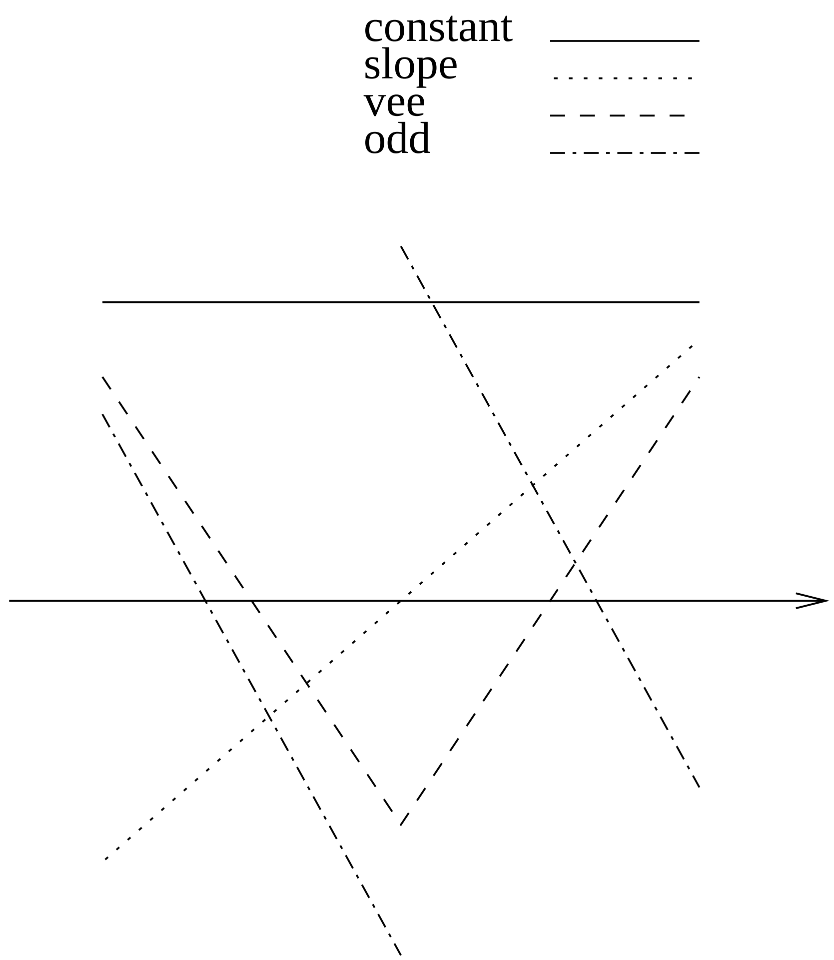
Same overall strategy:

- Find best "linear multiwavelet" representation
- Cull back to a robust representation, $w_{\text{rob}}$
- Output best $B$-bucket piecewise-linear representation to $w_{\text{rob}}$.

Next:

- What are linear multiwavelets?
- How to find best $B$-bucket piecewise-linear representation.

# Linear Multiwavelets, Graphical



# Linear Multiwavelets

E.g.,

$$\begin{array}{cccccccc}
+1 & +1 & +1 & +1 & +1 & +1 & +1 & +1 \\
-7 & -5 & -3 & -1 & +1 & +3 & +5 & +7 \\
+3 & +1 & -1 & -3 & -3 & -1 & +1 & +3 \\
+7 & -1 & -9 & -17 & +17 & +9 & +1 & -7 \\
+1 & -1 & -1 & +1 & 0 & 0 & 0 & 0 \\
-1 & +3 & -3 & +1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & +1 & -1 & -1 & +1 \\
0 & 0 & 0 & 0 & -1 & +3 & -3 & +1
\end{array}$$

# Linear Multiwavelets: Properties

- ONB
- Linear Multiwavelets and pw-linear representations simulate each other with $O(\log(d))$-factor blowup

# Best Piecewise-Linear Representation

Have $w_{\text{rob}}$ (pw-linear rep'n with $B' \approx B \cdot \log(d)/\epsilon$ pieces)

Want best $B$-bucket pw-linear repn $h$ to $w_{\text{rob}}$. Recall best 1-bucket repn to $x$ is

$$\langle x, \psi \rangle \, \psi + \langle x, \phi \rangle \, \phi,$$

where $\psi$ is constant and $\phi$ is slant.

Need

- New prefix arrays
- "Dual Dynamic Programming;" cost polynomial in $B \log(d)/\epsilon$.

# Prefix arrays:

- Get $\langle x, \psi \rangle$ from $Px$
- Get $\langle x, \phi \rangle$ from $P(x \cdot \phi)$ and $Px$
- Error of $a \cdot \psi + b \cdot \phi$ to $x$ is

$$\|x - (a \cdot \psi + b \cdot \phi)\|^2 = \langle x - (a \cdot \psi + b \cdot \phi), x - (a \cdot \psi + b \cdot \phi) \rangle .$$

Also need $P(x^2)$.

# Dual Dynamic Programming

Define $\mathrm{Far}[k,m]$ as the biggest $j$ such that there's a $k$-bucket histogram on $[0,j)$ with error at most $m$ (in appropriate units).

Assume we know $E$ with $\frac{1}{2}E \le E_{\mathrm{opt}} \le E$.

Consider $m = 0, \epsilon E/B, 2\epsilon E/B, \ldots, 2E$. ($B/\epsilon$ possibilities for $m$; coarse granularity leads to $\epsilon E/B$ extra error per boundary—$\epsilon E$ in all).

Thus: $\mathrm{Far}[k,m] = \max_n \{j : n + \mathrm{Cost}[\mathrm{Far}[k-1,n], j] < m\}$.

"Go as far as we can with $k-1$ buckets and error $n$, then add 1 bucket. Try all $n$."

Runtime: $k < B$, $m < B/\epsilon$, $n < B/\epsilon$, find $j$ by binary search: $O(B^3 \log(d)/\epsilon^2)$.

# Rangesum histograms

Given $x$, want pw-constant $h$ to optimize *range* queries to $x$:

$$\sum_{\ell,r}\left(\sum_{\ell\leq i<r} h - x_i\right)^2 .$$

Height $h$ of a bucket affects many non-local queries.

Foils previous tricks. Instead, transform to prefix domain.

# Transform to Prefix domain

$$\begin{aligned}
\sum_{\ell,r}\left(\sum_{\ell\le i<r} h_i - x_i\right)^2
&= \sum_{\ell,r}((P(h-x))_r - (P(h-x))_\ell)^2 \\
&= \sum_{\ell,r}(P(h-x))_r^2 + (P(h-x))_\ell^2 - 2P(h-x)_r P(h-x)_\ell \\
&= 2d\sum_{\ell}((Ph)_\ell - (Px)_\ell)^2 \quad \text{(we'll make } \textstyle\sum_\ell P(h-x)_\ell = 0.\text{)} \\
&= 2d\,\|Ph - Px\|^2,
\end{aligned}$$

Get *point*-query problem.

# Prefix array of histograms

If $h$ is pw-constant, then $Ph$ is piecewise-linear connected

Do not know how to find near-best pwlc approx to given $Px$ (equivalent to original problem).

Find near-best $B$-bucket pw-linear (disconnected) approx to $Px$ under point queries.

Leads to $(2B)$-bucket pw-constant repn for range queries to $x$.

# Simulate/Invert Prefix Array

When reading $x$, simulate reading $Px$:

- "add 5 to $x_3$" becomes "add 5 to $(Px)_3, (Px)_4, (Px)_5, \ldots$"
- Affects only $O(\log(d))$ linear multiwavelets (whose support includes 3).

From $Ph$, recover $h_i = (\Delta(Ph))_i = (Ph)_{i+1} - (Ph)_i$.

# Overall algorithm

- When reading $x$, simulate reading $Px$.
- Find best $(2B)$-bucket pw-linear approx $\ell$ to $Px$ under point queries
- Make sure $\text{avg}(\ell) = \text{avg}(Px)$. (Approximately enforced automatically by optimality.)
- Output $\Delta\ell$ as $(2, 1+\epsilon)$ approximation, i.e., $2B$ buckets, $(1+\epsilon)$ times best error under range queries.

# Overview of Summaries

- Heavy Hitters
- Weak greedy sparse recovery
- Orthonormal change of basis
- Haar Wavelets
- Histograms (piecewise constant)
- Multi-dimensional (hierarchical)
- Piecewise-linear
- Range queries