Adaptive Sparse Recovery with Limited Adaptivity

Akshay Kamath **Eric Price**

UT Austin

2018-11-27

Outline

Outline

Sparsity

- An n-dimensional vector x is "k-sparse" if only k non-zero coefficients.

Sparsity

- An n-dimensional vector x is "k-sparse" if only k non-zero coefficients.
- "Approximate sparsity:" vector "close" to a sparse vector

Sparsity

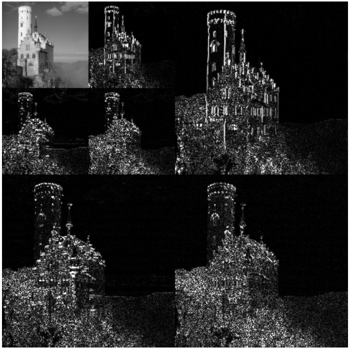
- An n-dimensional vector x is "k-sparse" if only k non-zero coefficients.
- "Approximate sparsity:" vector "close" to a sparse vector
- Approximate sparsity is a common form of structure.

Sparsity

- An n-dimensional vector x is "k-sparse" if only k non-zero coefficients.
- "Approximate sparsity:" vector "close" to a sparse vector
- Approximate sparsity is a common form of structure.

Sparsity

- An n-dimensional vector x is "k-sparse" if only k non-zero coefficients.
- "Approximate sparsity:" vector "close" to a sparse vector
- Approximate sparsity is a common form of structure.
- Images sparse in wavelet basis



Sparse Recovery / Compressive Sensing

AKA heavy hitters/frequency estimation in turnstile streams

Sparse Recovery / Compressive Sensing

AKA heavy hitters/frequency estimation in turnstile streams

- Suppose an n-dimensional vector x is k-sparse in known basis.

Sparse Recovery / Compressive Sensing

AKA heavy hitters/frequency estimation in turnstile streams

- Suppose an n-dimensional vector x is k-sparse in known basis.
- Observe Ax, a set of $m << n$ linear products.

Sparse Recovery / Compressive Sensing

AKA heavy hitters/frequency estimation in turnstile streams

- Suppose an n-dimensional vector x is k-sparse in known basis.
- Observe Ax, a set of $m << n$ linear products.
- Why linear? Many applications:

Sparse Recovery / Compressive Sensing

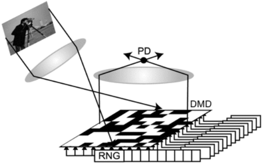
AKA heavy hitters/frequency estimation in turnstile streams

- Suppose an n-dimensional vector x is k-sparse in known basis.
- Observe Ax, a set of $m << n$ linear products.
- Why linear? Many applications:
  - Genetic testing: mixing blood samples.

Sparse Recovery / Compressive Sensing

AKA heavy hitters/frequency estimation in turnstile streams

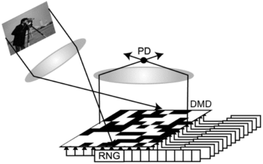
- Suppose an n-dimensional vector x is k-sparse in known basis.
- Observe Ax, a set of $m << n$ linear products.
- Why linear? Many applications:
  - Genetic testing: mixing blood samples.
  - Streaming updates: $A(x + \Delta) = Ax + A\Delta$.



Sparse Recovery / Compressive Sensing

AKA heavy hitters/frequency estimation in turnstile streams

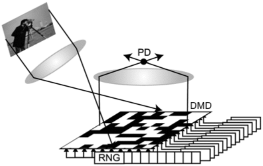
- Suppose an n-dimensional vector x is k-sparse in known basis.
- Observe Ax, a set of $m << n$ linear products.
- Why linear? Many applications:
  - Genetic testing: mixing blood samples.
  - Streaming updates: $A(x + \Delta) = Ax + A\Delta$.
  - Camera optics: filter in front of lens.



Sparse Recovery / Compressive Sensing

AKA heavy hitters/frequency estimation in turnstile streams

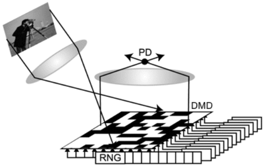
- Suppose an n-dimensional vector x is k-sparse in known basis.
- Observe Ax, a set of $m << n$ linear products.
- Why linear? Many applications:
  - Genetic testing: mixing blood samples.
  - Streaming updates: $A(x + \Delta) = Ax + A\Delta$.
  - Camera optics: filter in front of lens.
- Goal is to robustly recover x from Ax.
  - Informally: get close to x if x is close to k-sparse.



Sparse Recovery / Compressive Sensing

AKA heavy hitters/frequency estimation in turnstile streams

- Suppose an n-dimensional vector x is k-sparse in known basis.
- Observe Ax, a set of $m << n$ linear products.
- Why linear? Many applications:
  - Genetic testing: mixing blood samples.
  - Streaming updates: $A(x + \Delta) = Ax + A\Delta$.
  - Camera optics: filter in front of lens.
- Goal is to robustly recover x from Ax.
  - Informally: get close to x if x is close to k-sparse.
- Extremely well studied: thousands of papers.



Standard Sparse Recovery Framework

- Specify distribution on $m \times n$ matrices A (independent of x).

- Given linear sketch Ax, recover $\widehat{x}$.
- Satisfying the recovery guarantee:

$$\|\widehat{x} - x\|_2 \leqslant C \min_{k\text{-sparse } x_k} \|x - x_k\|_2$$

with probability $2/3$.

Standard Sparse Recovery Framework

- Specify distribution on $m \times n$ matrices A (independent of x).

- Given linear sketch Ax, recover $\widehat{x}$.
- Satisfying the recovery guarantee:

$$\|\widehat{x} - x\|_2 \leqslant C \min_{k\text{-sparse } x_k} \|x - x_k\|_2$$

  with probability $2/3$.
- Solvable with $\Theta(k \log \frac{n}{k})$ measurements [Càndes-Romberg-Tao '06].

Standard Sparse Recovery Framework

- Specify distribution on $m \times n$ matrices A (independent of x).
  - Choose matrix A_i based on previous observations (possibly randomized).
  - Observe $A_i x$.
  - Number of measurements m is total number of rows in all A_i.
  - Number of *rounds* is R.
- Given linear sketch Ax, recover $\widehat{x}$.
- Satisfying the recovery guarantee:

$$\|\widehat{x} - x\|_2 \leqslant C \min_{k\text{-sparse } x_k} \|x - x_k\|_2$$

  with probability 2/3.
- Solvable with $\Theta(k \log \frac{n}{k})$ measurements [Càndes-Romberg-Tao '06].

Standard Sparse Recovery Framework

- Specify distribution on $m \times n$ matrices A (independent of x).
  - Choose matrix A_i based on previous observations (possibly randomized).
  - Observe $A_i x$.
  - Number of measurements m is total number of rows in all A_i.
  - Number of *rounds* is R.
- Given linear sketch Ax, recover $\widehat{x}$.
- Satisfying the recovery guarantee:

$$\|\widehat{x} - x\|_2 \leqslant C \min_{k\text{-sparse } x_k} \|x - x_k\|_2$$

  with probability 2/3.
- Solvable with $\Theta(k \log \frac{n}{k})$ measurements [Càndes-Romberg-Tao '06].
- Solvable in $O(k \log \log \frac{n}{k})$ [Indyk-Price-Woodruff '11].

Prior Work

- Nonadaptively: $m \approx \frac{1}{\varepsilon} k \log n$ for $C = 1 + \varepsilon$.

Prior Work

- Nonadaptively: $m \approx \frac{1}{\varepsilon} k \log n$ for $C = 1 + \varepsilon$.
- One line of work: $\varepsilon = o(1)$ for $m \approx k \log n$.
  - [Malioutov, Sanghavi, Willski '08], [Castro, Haupt, Nowak, Raz '08], [Haupt, Castro, Nowak '11], [Haupt, Baraniuk, Castro, Nowak '12]

Prior Work

- Nonadaptively: $m \approx \frac{1}{\varepsilon} k \log n$ for $C = 1 + \varepsilon$.
- One line of work: $\varepsilon = o(1)$ for $m \approx k \log n$.
  - [Malioutov, Sanghavi, Willski '08], [Castro, Haupt, Nowak, Raz '08], [Haupt, Castro, Nowak '11], [Haupt, Baraniuk, Castro, Nowak '12]
- Another line: also allows $m \ll k \log n$.
  - [Indyk-Price-Woodruff '11], [Nakos, Shi, Woodruff, Zhang '18]

$$m \lesssim \frac{\log \log(1/\varepsilon)}{\varepsilon} k + k \log \log n$$

Prior Work

- Nonadaptively: $m \approx \frac{1}{\varepsilon} k \log n$ for $C = 1 + \varepsilon$.
- One line of work: $\varepsilon = o(1)$ for $m \approx k \log n$.
  - [Malioutov, Sanghavi, Willski '08], [Castro, Haupt, Nowak, Raz '08], [Haupt, Castro, Nowak '11], [Haupt, Baraniuk, Castro, Nowak '12]
- Another line: also allows $m \ll k \log n$.
  - [Indyk-Price-Woodruff '11], [Nakos, Shi, Woodruff, Zhang '18]

$$m \lesssim \frac{\log \log(1/\varepsilon)}{\varepsilon} k + k \log \log n$$

- Lower bounds:
  - [Arias-Castro, Candès, Davenport '13]: $m \gtrsim \frac{1}{\varepsilon} k$
  - [Price, Woodruff '13]: $m \gtrsim \log \log n$.

Results in adaptive sparse recovery, $C = O(1)$

- Unlimited adaptivity: with unlimited rounds,

$$k + \log \log n \lesssim m^* \lesssim k \cdot \log \log n$$

Results in adaptive sparse recovery, $C = O(1)$

- Unlimited adaptivity: with unlimited rounds,

$$k + \log\log n \lesssim m^* \lesssim k \cdot \log\log n$$

- Limited adaptivity: with $R = O(1)$ rounds,

$$k + \log^{1/R} n \lesssim m^* \lesssim k \cdot \log^{1/(R-3)} n.$$

Results in adaptive sparse recovery, $C = O(1)$

- Unlimited adaptivity: with unlimited rounds,

$$k + \log\log n \lesssim m^* \lesssim k \cdot \log\log n$$

- Limited adaptivity: with $R = O(1)$ rounds,

$$k + \log^{1/R} n \lesssim m^* \lesssim k \cdot \log^{1/(R-3)} n.$$

- New results: with $R = O(1)$ rounds,

$$k \cdot \log^{1/R} n \lesssim m^* \lesssim k \cdot \log^{1/R} n \log^* k$$

With caveat:

Results in adaptive sparse recovery, $C = O(1)$

- Unlimited adaptivity: with unlimited rounds,

$$k + \log\log n \lesssim m^* \lesssim k \cdot \log\log n$$

- Limited adaptivity: with $R = O(1)$ rounds,

$$k + \log^{1/R} n \lesssim m^* \lesssim k \cdot \log^{1/(R-3)} n.$$

- New results: with $R = O(1)$ rounds,

$$k \cdot \log^{1/R} n \lesssim m^* \lesssim k \cdot \log^{1/R} n \log^* k$$

With caveat: the lower bound only applies for $k < 2^{\log^{1/R} n}$ $\iff m^* > k \log k$.

Results in adaptive sparse recovery, $C = O(1)$

- Unlimited adaptivity: with unlimited rounds,

$$k + \log\log n \lesssim m^* \lesssim k \cdot \log\log n$$

- Limited adaptivity: with $R = O(1)$ rounds,

$$k + \log^{1/R} n \lesssim m^* \lesssim k \cdot \log^{1/(R-3)} n.$$

- New results: with $R = O(1)$ rounds,

$$k \cdot \log^{1/R} n \lesssim m^* \lesssim k \cdot \log^{1/R} n \log^* k$$

With caveat: the lower bound only applies for $k < 2^{\log^{1/R} n}$
$\iff m^* > k \log k$.
For $k < n^{o(1)}$, $m^* = \omega(k)$.

Outline

Well-understood setting: $k = 1$

Theorem (Indyk-Price-Woodruff '11, Price-Woodruff '13)

R-round 1-sparse recovery requires $\Theta(R \log^{1/R} n)$ measurements.

Well-understood setting: $k = 1$

Theorem (Indyk-Price-Woodruff '11, Price-Woodruff '13)

R-round 1-sparse recovery requires $\Theta(R \log^{1/R} n)$ measurements.

- Outline of this section:

Well-understood setting: $k = 1$

Theorem (Indyk-Price-Woodruff '11, Price-Woodruff '13)

R-round 1-sparse recovery requires $\Theta(R \log^{1/R} n)$ measurements.

- Outline of this section:
  - $R = 1$ lower bound: $\Omega(\log n)$.

Well-understood setting: $k = 1$

Theorem (Indyk-Price-Woodruff '11, Price-Woodruff '13)

R-round 1-sparse recovery requires $\Theta(R \log^{1/R} n)$ measurements.

- Outline of this section:
  - $R = 1$ lower bound: $\Omega(\log n)$.
  - Adaptive upper bound: $O(\log \log n)$.

Well-understood setting: $k = 1$

Theorem (Indyk-Price-Woodruff '11, Price-Woodruff '13)

R-round 1-sparse recovery requires $\Theta(R \log^{1/R} n)$ measurements.

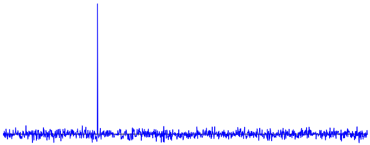
- Outline of this section:
  - $R = 1$ lower bound: $\Omega(\log n)$.
  - Adaptive upper bound: $O(\log \log n)$.
  - Adaptive lower bound: $\Omega(\log \log n)$.

Well-understood setting: $k = 1$

Theorem (Indyk-Price-Woodruff '11, Price-Woodruff '13)

R-round 1-sparse recovery requires $\Theta(R \log^{1/R} n)$ measurements.

- Outline of this section:
  - $R = 1$ lower bound: $\Omega(\log n)$.
  - Adaptive upper bound: $O(\log \log n)$.
  - Adaptive lower bound: $\Omega(\log \log n)$.
- Hard case: x is random e_z plus Gaussian noise w with $\|w\|_2 \approx 1$.

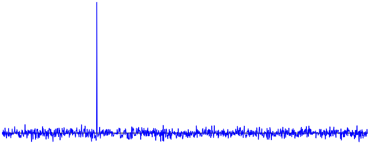


Well-understood setting: $k = 1$

Theorem (Indyk-Price-Woodruff '11, Price-Woodruff '13)

R-round 1-sparse recovery requires $\Theta(R \log^{1/R} n)$ measurements.

- Outline of this section:
  - $R = 1$ lower bound: $\Omega(\log n)$.
  - Adaptive upper bound: $O(\log \log n)$.
  - Adaptive lower bound: $\Omega(\log \log n)$.
- Hard case: x is random e_z plus Gaussian noise w with $\|w\|_2 \approx 1$.



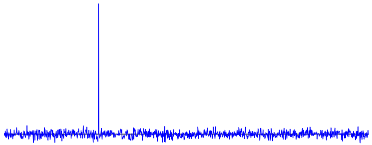
- Robust recovery must locate z.

Well-understood setting: $k = 1$

Theorem (Indyk-Price-Woodruff '11, Price-Woodruff '13)

R-round 1-sparse recovery requires $\Theta(R \log^{1/R} n)$ *measurements.*

- Outline of this section:
  - $R = 1$ lower bound: $\Omega(\log n)$.
  - Adaptive upper bound: $O(\log \log n)$.
  - Adaptive lower bound: $\Omega(\log \log n)$.
- Hard case: x is random e_z plus Gaussian noise w with $\|w\|_2 \approx 1$.



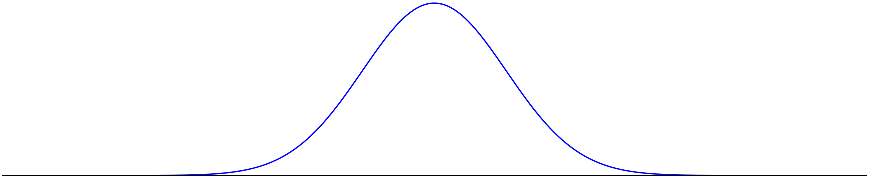
- Robust recovery must locate z.
- Observations $\langle v, x \rangle = v_z + \langle v, w \rangle = v_z + \frac{\|v\|_2}{\sqrt{n}} z$, for $z \sim N(0, 1)$.

1-sparse recovery: non-adaptive lower bound

- Observe $\langle v, x \rangle = v_z + \frac{\|v\|_2}{\sqrt{n}} z$, where $z \sim N(0, \Theta(1))$

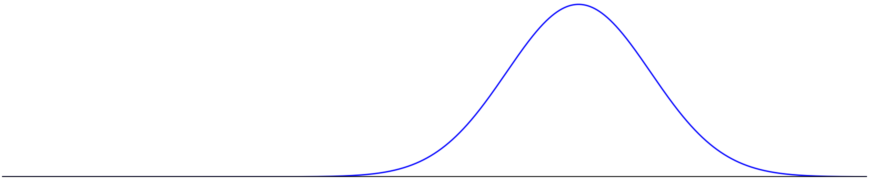
1-sparse recovery: non-adaptive lower bound

- Observe $\langle v, x \rangle = v_z + \frac{\|v\|_2}{\sqrt{n}} z$, where $z \sim N(0, \Theta(1))$



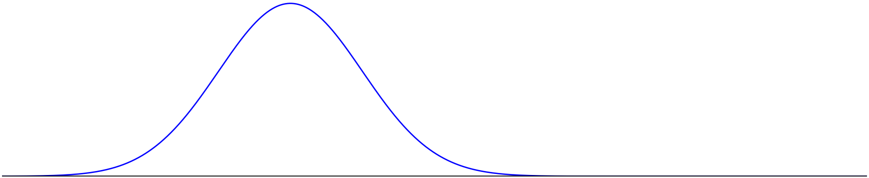
1-sparse recovery: non-adaptive lower bound

- Observe $\langle v, x \rangle = v_z + \frac{\|v\|_2}{\sqrt{n}} z$, where $z \sim N(0, \Theta(1))$



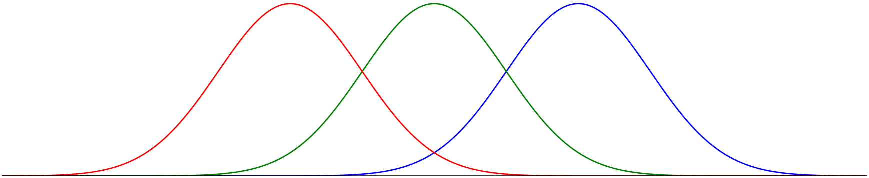
1-sparse recovery: non-adaptive lower bound

- Observe $\langle v, x \rangle = v_z + \frac{\|v\|_2}{\sqrt{n}} z$, where $z \sim N(0, \Theta(1))$



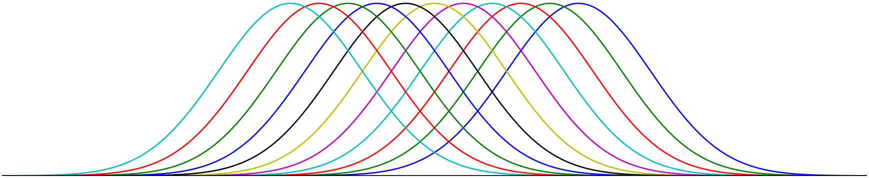
1-sparse recovery: non-adaptive lower bound

- Observe $\langle v, x \rangle = v_z + \frac{\|v\|_2}{\sqrt{n}} z$, where $z \sim N(0, \Theta(1))$



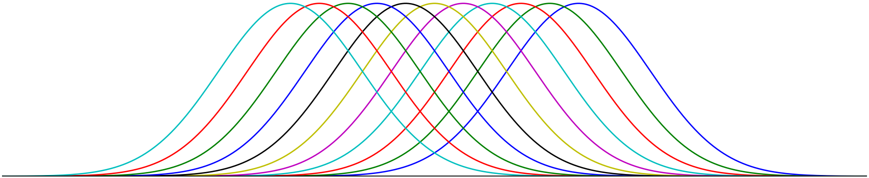
1-sparse recovery: non-adaptive lower bound

- Observe $\langle v, x \rangle = v_z + \frac{\|v\|_2}{\sqrt{n}} z$, where $z \sim N(0, \Theta(1))$



1-sparse recovery: non-adaptive lower bound

- Observe $\langle v, x \rangle = v_z + \frac{\|v\|_2}{\sqrt{n}} z$, where $z \sim N(0, \Theta(1))$



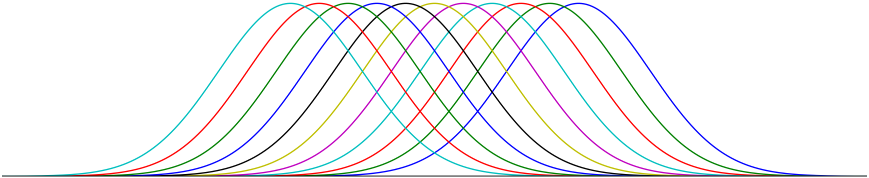
- Shannon-Hartley theorem: AWGN channel capacity is

$$I(z, \langle v, x \rangle) \leqslant \frac{1}{2} \log(1 + \mathrm{SNR})$$

where *SNR* denotes the "signal-to-noise ratio,"

1-sparse recovery: non-adaptive lower bound

- Observe $\langle v, x\rangle = v_z + \frac{\|v\|_2}{\sqrt{n}} z$, where $z \sim N(0, \Theta(1))$



- Shannon-Hartley theorem: AWGN channel capacity is

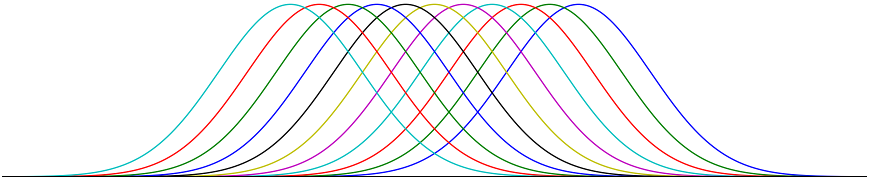
$$I(z, \langle v, x\rangle) \leqslant \frac{1}{2}\log(1 + \mathrm{SNR})$$

where *SNR* denotes the "signal-to-noise ratio,"

$$SNR = \frac{\mathbb{E}[\mathrm{signal}^2]}{\mathbb{E}[\mathrm{noise}^2]} \eqsim \frac{\mathbb{E}[v_z^2]}{\|v\|_2^2/n}$$

1-sparse recovery: non-adaptive lower bound

- Observe $\langle v, x \rangle = v_z + \frac{\|v\|_2}{\sqrt{n}} z$, where $z \sim N(0, \Theta(1))$



- Shannon-Hartley theorem: AWGN channel capacity is

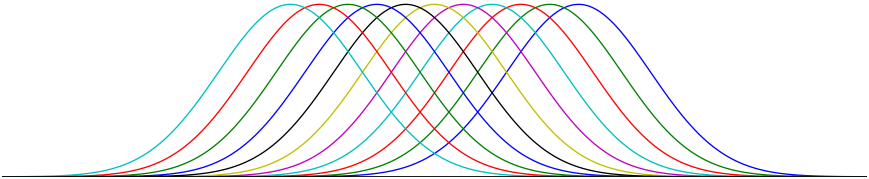
$$I(z, \langle v, x \rangle) \leqslant \frac{1}{2} \log(1 + \mathrm{SNR})$$

where *SNR* denotes the "signal-to-noise ratio,"

$$SNR = \frac{\mathbb{E}[\text{signal}^2]}{\mathbb{E}[\text{noise}^2]} \eqsim \frac{\mathbb{E}[v_z^2]}{\|v\|_2^2/n} = 1$$

1-sparse recovery: non-adaptive lower bound

- Observe $\langle v, x \rangle = v_z + \frac{\|v\|_2}{\sqrt{n}} z$, where $z \sim N(0, \Theta(1))$



- Shannon-Hartley theorem: AWGN channel capacity is

$$I(z, \langle v, x \rangle) \leqslant \frac{1}{2} \log(1 + \mathrm{SNR})$$

where *SNR* denotes the "signal-to-noise ratio,"

$$SNR = \frac{\mathbb{E}[\text{signal}^2]}{\mathbb{E}[\text{noise}^2]} \eqsim \frac{\mathbb{E}[v_z^2]}{\|v\|_2^2/n} = 1$$

- Finding z needs $\Omega(\log n)$ non-adaptive measurements.

1-sparse recovery: changes in adaptive setting

- Information capacity

$$I(z, \langle v, x\rangle) \leqslant \frac{1}{2}\log(1+\mathrm{SNR}).$$

where SNR denotes the "signal-to-noise ratio,"

$$SNR \approx \frac{\mathbb{E}[v_z^2]}{\|v\|_2^2/n}.$$

1-sparse recovery: changes in adaptive setting

- Information capacity

$$I(z, \langle v, x \rangle) \leqslant \frac{1}{2} \log(1 + \text{SNR}).$$

where SNR denotes the "signal-to-noise ratio,"

$$SNR \approx \frac{\mathbb{E}[v_z^2]}{\|v\|_2^2 / n}.$$

- If z is independent of v, this is 1.

1-sparse recovery: changes in adaptive setting

- Information capacity

$$I(z, \langle v, x\rangle) \leqslant \frac{1}{2}\log(1 + \mathrm{SNR}).$$

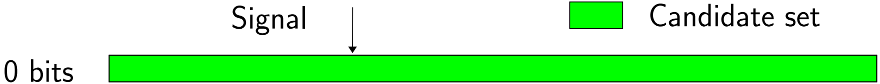
where *SNR* denotes the "signal-to-noise ratio,"

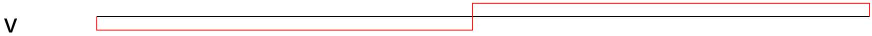
$$SNR \approx \frac{\mathbb{E}[v_z^2]}{\|v\|_2^2/n}.$$

- If z is independent of v, this is 1.
- As we learn about z, we can increase the SNR.

1-sparse recovery: adaptive upper bound

$$x = e_z + w$$

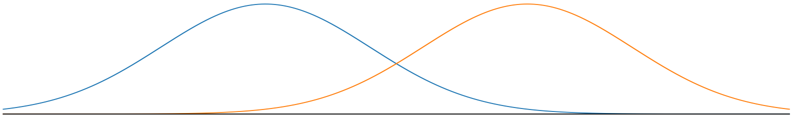




$SNR = 2$

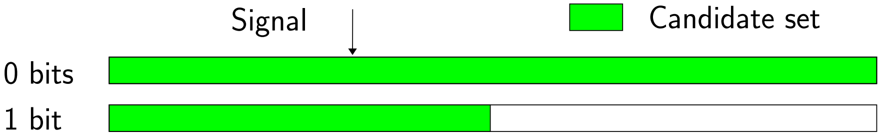
$I(z, \langle v, x\rangle) \leqslant \log SNR = 1$

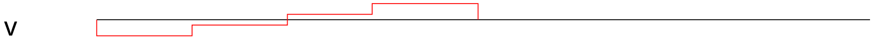
$$\langle v, x\rangle = v_z + \langle v, w\rangle$$



1-sparse recovery: adaptive upper bound

$$x = e_z + w$$

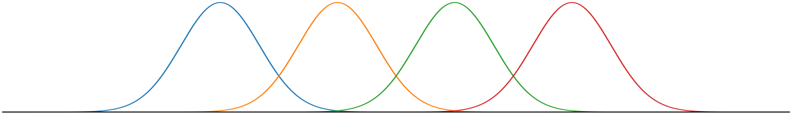




$SNR = 2^2$

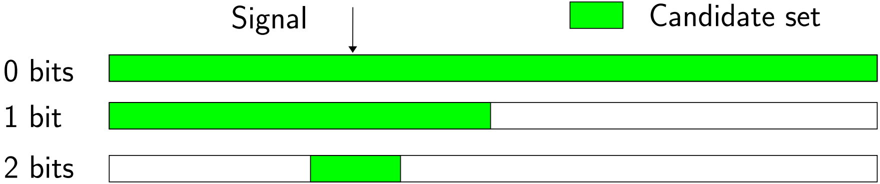
$I(z, \langle v, x \rangle) \leqslant \log SNR = 2$

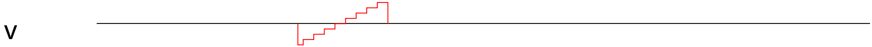
$$\langle v, x \rangle = v_z + \langle v, w \rangle$$



1-sparse recovery: adaptive upper bound

$x = e_z + w$

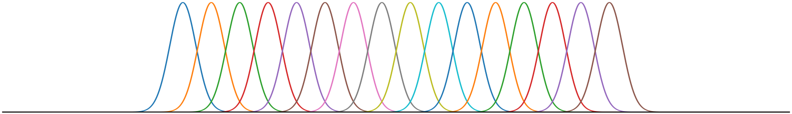




$SNR = 2^4$

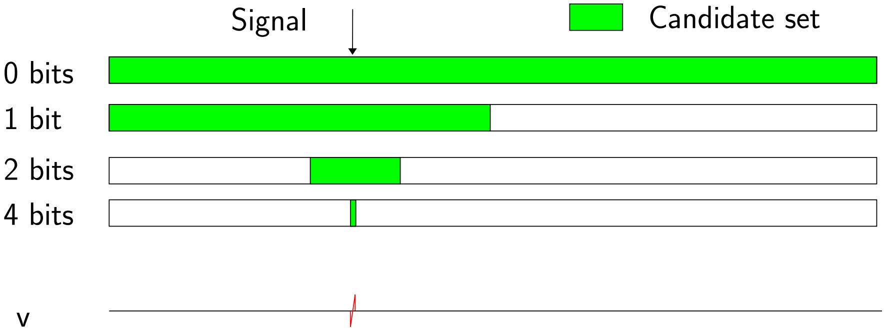
$I(z, \langle v, x \rangle) \leqslant \log SNR = 4$

$\langle v, x \rangle = v_z + \langle v, w \rangle$



1-sparse recovery: adaptive upper bound

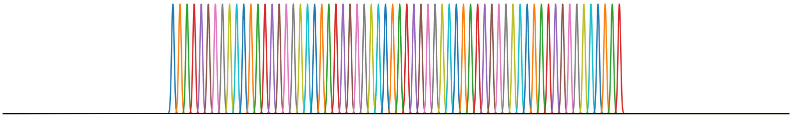
$$x = e_z + w$$



$SNR = 2^8$

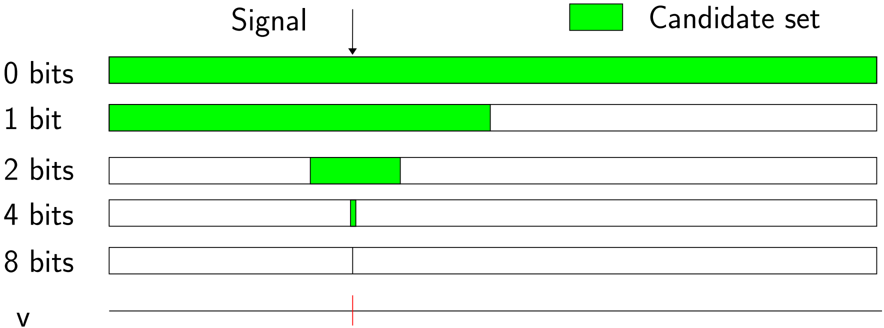
$I(z, \langle v, x \rangle) \leqslant \log SNR = 8$

$$\langle v, x \rangle = v_z + \langle v, w \rangle$$



1-sparse recovery: adaptive upper bound

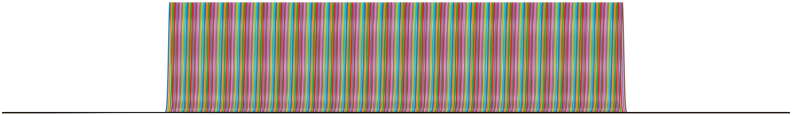
$$x = e_z + w$$



$SNR = 2^{16}$

$I(z, \langle v, x \rangle) \leqslant \log SNR = 16$

$$\langle v, x \rangle = v_z + \langle v, w \rangle$$



1-sparse recovery: adaptive lower bound

- Review of upper bound:
  - Given b bits of information about z.
  - Identifies z to set of size $n/2^b$.
  - Increases SNR, $\mathbb{E}[v_z^2]$, by 2^b.
  - Recover b bits of information in one measurement.
  - $1 \to 2 \to \cdots \to \log n$ in $\log \log n$ measurements.
  - $R = 2$: $1 \to \sqrt{\log n} \to \log n$ in $\sqrt{\log n}$ measurements/round.

1-sparse recovery: adaptive lower bound

- Review of upper bound:
  - Given b bits of information about z.
  - Identifies z to set of size $n/2^b$.
  - Increases SNR, $\mathbb{E}[v_z^2]$, by 2^b.
  - Recover b bits of information in one measurement.
  - $1 \to 2 \to \cdots \to \log n$ in $\log \log n$ measurements.
  - $R = 2$: $1 \to \sqrt{\log n} \to \log n$ in $\sqrt{\log n}$ measurements/round.
- Lower bound outline:
  - At each stage, have posterior distribution p on z.
  - $b = \log n - H(p)$ bits known.

1-sparse recovery: adaptive lower bound

- Review of upper bound:
  - Given b bits of information about z.
  - Identifies z to set of size $n/2^b$.
  - Increases SNR, $\mathbb{E}[v_z^2]$, by 2^b.
  - Recover b bits of information in one measurement.
  - $1 \to 2 \to \cdots \to \log n$ in $\log \log n$ measurements.
  - $R = 2$: $1 \to \sqrt{\log n} \to \log n$ in $\sqrt{\log n}$ measurements/round.
- Lower bound outline:
  - At each stage, have posterior distribution p on z.
  - $b = \log n - H(p)$ bits known.

Lemma (Key lemma for $k = 1$)

For any measurement vector v,

$$I(z; \langle v, x \rangle) \lesssim b + 1$$

1-sparse recovery: adaptive lower bound

- Lower bound outline:
  - At each stage, have posterior distribution p on z.
  - $b = \log n - H(p)$ bits known.
  - Show any measurement gives $O(b+1)$ bits of information.

1-sparse recovery: adaptive lower bound

- Lower bound outline:
  - At each stage, have posterior distribution p on z.
  - $b = \log n - H(p)$ bits known.
  - Show any measurement gives $O(b+1)$ bits of information.
- Shannon-Hartley:

$$I(z; \langle v, x \rangle) \leqslant \frac{1}{2} \log(1 + SNR) \lesssim 1 + \log \frac{\sum v_z^2 p_z}{\sum v_z^2 / n}$$

1-sparse recovery: adaptive lower bound

- Lower bound outline:
  - At each stage, have posterior distribution p on z.
  - $b = \log n - H(p)$ bits known.
  - Show any measurement gives $O(b+1)$ bits of information.
- Shannon-Hartley:

$$I(z; \langle v, x \rangle) \leqslant \frac{1}{2} \log(1 + SNR) \lesssim 1 + \log \frac{\sum v_z^2 p_z}{\sum v_z^2 / n} \lesssim 1 + n\|p\|_\infty$$

1-sparse recovery: adaptive lower bound

- Lower bound outline:
  - At each stage, have posterior distribution p on z.
  - $b = \log n - H(p)$ bits known.
  - Show any measurement gives $O(b+1)$ bits of information.
- Shannon-Hartley:

$$I(z; \langle v, x \rangle) \leqslant \frac{1}{2}\log(1 + SNR) \lesssim 1 + \log \frac{\sum v_z^2 p_z}{\sum v_z^2/n} \lesssim 1 + n\|p\|_\infty$$

- Bound is good ($SNR \approx 2^b$) when nonzero p_z are similar.

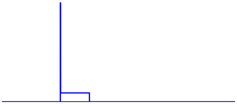
1-sparse recovery: adaptive lower bound

- Lower bound outline:
  - At each stage, have posterior distribution p on z.
  - $b = \log n - H(p)$ bits known.
  - Show any measurement gives $O(b+1)$ bits of information.
- Shannon-Hartley:

$$I(z; \langle v, x\rangle) \leqslant \frac{1}{2}\log(1+SNR) \lesssim 1 + \log \frac{\sum v_z^2 p_z}{\sum v_z^2/n} \lesssim 1 + n\|p\|_\infty$$

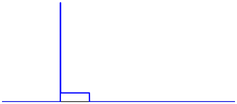
- Bound is good ($SNR \approx 2^b$) when nonzero p_z are similar.
- Can be terrible in general: $b = 1$ but $SNR = n/\log n$.

1-sparse recovery: adaptive lower bound



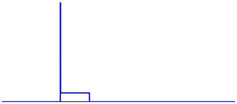
- Lower bound outline:
  - At each stage, have posterior distribution p on z.
  - $b = \log n - H(p) = \sum p_z \log np_z$ bits known.
  - Show any measurement gives $O(b + 1)$ bits of information.

1-sparse recovery: adaptive lower bound



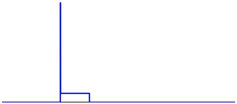
- Lower bound outline:
  - At each stage, have posterior distribution p on z.
  - $b = \log n - H(p) = \sum p_z \log np_z$ bits known.
  - Show any measurement gives $O(b + 1)$ bits of information.
- Partition indices into "level sets" $S_0, S_1, \ldots \subseteq [n]$ of p:

1-sparse recovery: adaptive lower bound



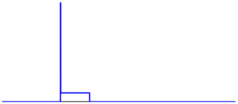
- Lower bound outline:
  - At each stage, have posterior distribution p on z.
  - $b = \log n - H(p) = \sum p_z \log np_z$ bits known.
  - Show any measurement gives $O(b+1)$ bits of information.
- Partition indices into "level sets" $S_0, S_1, \ldots \subseteq [n]$ of p:
  - $S_J = \{z \mid p_z \in [2^J/n, 2^{J+1}/n]\}$

1-sparse recovery: adaptive lower bound



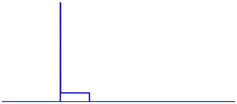
- Lower bound outline:
  - At each stage, have posterior distribution p on z.
  - $b = \log n - H(p) = \sum p_z \log np_z$ bits known.
  - Show any measurement gives $O(b+1)$ bits of information.
- Partition indices into "level sets" $S_0, S_1, \ldots \subseteq [n]$ of p:
  - $S_J = \{z \mid p_z \in [2^J/n, 2^{J+1}/n]\}$
  - $\mathbb{E}[J] \leqslant b$.

1-sparse recovery: adaptive lower bound



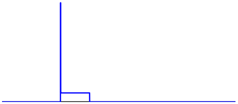
- Lower bound outline:
  - At each stage, have posterior distribution p on z.
  - $b = \log n - H(p) = \sum p_z \log np_z$ bits known.
  - Show any measurement gives $O(b+1)$ bits of information.
- Partition indices into "level sets" $S_0, S_1, \ldots \subseteq [n]$ of p:
  - $S_J = \{z \mid p_z \in [2^J/n, 2^{J+1}/n]\}$
  - $\mathbb{E}[J] \leqslant b$.
- $I(z; \langle v, x \rangle) \leqslant I(z; \langle v, x \rangle \mid J) + H(J)$.

1-sparse recovery: adaptive lower bound



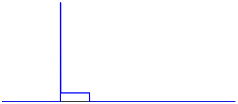
- Lower bound outline:
  - At each stage, have posterior distribution p on z.
  - $b = \log n - H(p) = \sum p_z \log np_z$ bits known.
  - Show any measurement gives $O(b+1)$ bits of information.
- Partition indices into "level sets" $S_0, S_1, \ldots \subseteq [n]$ of p:
  - $S_J = \{z \mid p_z \in [2^J/n, 2^{J+1}/n]\}$
  - $\mathbb{E}[J] \leqslant b$.
- $I(z; \langle v, x \rangle) \leqslant I(z; \langle v, x \rangle \mid J) + H(J)$.
- Shannon-Hartley: $I(z; \langle v, x \rangle \mid J = j) \lesssim j + 1$.

1-sparse recovery: adaptive lower bound



- Lower bound outline:
  - At each stage, have posterior distribution p on z.
  - $b = \log n - H(p) = \sum p_z \log np_z$ bits known.
  - Show any measurement gives $O(b+1)$ bits of information.
- Partition indices into "level sets" $S_0, S_1, \ldots \subseteq [n]$ of p:
  - $S_J = \{z \mid p_z \in [2^J/n, 2^{J+1}/n]\}$
  - $\mathbb{E}[J] \leqslant b$.
- $I(z; \langle v, x \rangle) \leqslant I(z; \langle v, x \rangle \mid J) + H(J)$.
- Shannon-Hartley: $I(z; \langle v, x \rangle \mid J = j) \lesssim j + 1$.

1-sparse recovery: adaptive lower bound



- Lower bound outline:
  - At each stage, have posterior distribution p on z.
  - $b = \log n - H(p) = \sum p_z \log np_z$ bits known.
  - Show any measurement gives $O(b+1)$ bits of information.
- Partition indices into "level sets" $S_0, S_1, \ldots \subseteq [n]$ of p:
  - $S_J = \{z \mid p_z \in [2^J/n, 2^{J+1}/n]\}$
  - $\mathbb{E}[J] \leqslant b$.
- $I(z; \langle v, x\rangle) \leqslant I(z; \langle v, x\rangle \mid J) + H(J)$.
- Shannon-Hartley: $I(z; \langle v, x\rangle \mid J = j) \lesssim j + 1$.

Lemma (Key lemma for $k = 1$)

$$I(z; \langle v, x\rangle) \lesssim b + 1$$

1-sparse recovery: adaptive lower bound: finishing up

Lemma (Key lemma for $k = 1$)

$$I(z; \langle v, x \rangle) \lesssim b + 1$$

1-sparse recovery: adaptive lower bound: finishing up

Lemma (Key lemma for $k = 1$)

$$I(z; \langle v, x \rangle) \lesssim b + 1$$

- Suppose two rounds with m measurements each.

1-sparse recovery: adaptive lower bound: finishing up

Lemma (Key lemma for $k = 1$)

$$I(z; \langle v, x \rangle) \lesssim b + 1$$

- Suppose two rounds with m measurements each.
  - $O(m)$ bits learned in first round.

1-sparse recovery: adaptive lower bound: finishing up

Lemma (Key lemma for $k = 1$)

$$I(z; \langle v, x \rangle) \lesssim b + 1$$

- Suppose two rounds with m measurements each.
  - $O(m)$ bits learned in first round.
  - $O(m^2)$ bits in second round.

1-sparse recovery: adaptive lower bound: finishing up

Lemma (Key lemma for $k = 1$)

$$I(z; \langle v, x \rangle) \lesssim b + 1$$

- Suppose two rounds with m measurements each.
  - $O(m)$ bits learned in first round.
  - $O(m^2)$ bits in second round.
  - Hence $m \gtrsim \sqrt{\log n}$.

1-sparse recovery: adaptive lower bound: finishing up

Lemma (Key lemma for $k = 1$)

$$I(z; \langle v, x \rangle) \lesssim b + 1$$

- Suppose two rounds with m measurements each.
  - $O(m)$ bits learned in first round.
  - $O(m^2)$ bits in second round.
  - Hence $m \gtrsim \sqrt{\log n}$.
- In general: $\Omega(R \log^{1/R} n)$ bits

1-sparse recovery: adaptive lower bound: finishing up

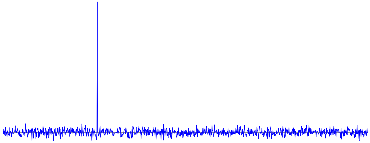
Lemma (Key lemma for $k = 1$)

$$I(z; \langle v, x \rangle) \lesssim b + 1$$

- Suppose two rounds with m measurements each.
  - $O(m)$ bits learned in first round.
  - $O(m^2)$ bits in second round.
  - Hence $m \gtrsim \sqrt{\log n}$.
- In general: $\Omega(R \log^{1/R} n)$ bits
  - $\Omega(\log \log n)$ for unlimited R.

Outline

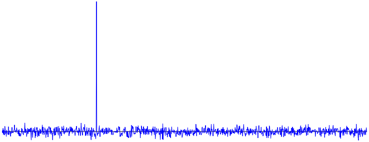
Recall the $k = 1$ proof outline



- Setting: $x = e_z + w$ for $z \sim p$.
- p is posterior on z from previous measurements.
- Previous measurements had information content

$$b := \log n - H(p)$$

Recall the $k = 1$ proof outline



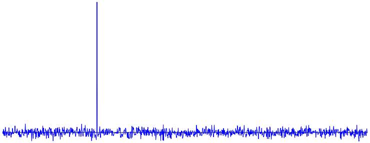
- Setting: $x = e_z + w$ for $z \sim p$.
- p is posterior on z from previous measurements.
- Previous measurements had information content

$$b := \log n - H(p)$$

Lemma (Key lemma for $k = 1$)

$$I(z; \langle v, x \rangle) \lesssim b + 1$$

Recall the $k = 1$ proof outline



- Setting: $x = e_z + w$ for $z \sim p$.
- p is posterior on z from previous measurements.
- Previous measurements had information content

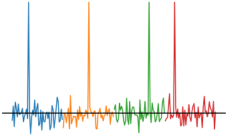
$$b := \log n - H(p)$$

Lemma (Key lemma for $k = 1$)

$$I(z; \langle v, x \rangle) \lesssim b + 1$$

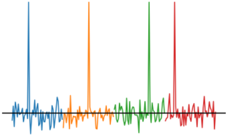
- **Question**: How to extend this to $k > 1$?

Extending to general k



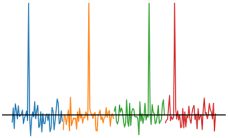
- Create k independent copies over domain $N = nk$.

Extending to general k



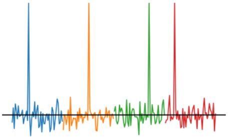
- Create k independent copies over domain $N = nk$.
- Formally: $x = \sum_{i=1}^{k} e_{ni+Z_i} + w$ for $Z \in [n]^k$, $Z \sim p$.

Extending to general k



- Create k independent copies over domain $N = nk$.
- Formally: $x = \sum_{i=1}^{k} e_{ni+Z_i} + w$ for $Z \in [n]^k$, $Z \sim p$.
- p is posterior from previous measurements.

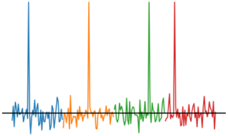
Extending to general k



- Create k independent copies over domain $N = nk$.
- Formally: $x = \sum_{i=1}^{k} e_{ni+Z_i} + w$ for $Z \in [n]^k$, $Z \sim p$.
- p is posterior from previous measurements.
- Previous measurements have information content

$$b := k \log n - H(p)$$

Extending to general k



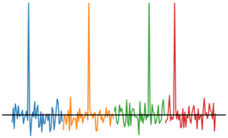
- Create k independent copies over domain $N = nk$.
- Formally: $x = \sum_{i=1}^{k} e_{ni+Z_i} + w$ for $Z \in [n]^k$, $Z \sim p$.
- p is posterior from previous measurements.
- Previous measurements have information content

$$b := k \log n - H(p)$$

Lemma (Key lemma for general k)

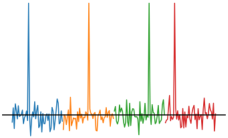
$$\sout{I(Z; \langle v, x \rangle) \lesssim b + 1} \text{ ????}$$

What lemma do we want for general k?



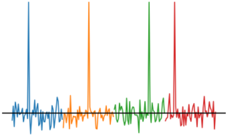
- $I(Z; \langle v, x \rangle) \lesssim b + 1$

What lemma do we want for general k?



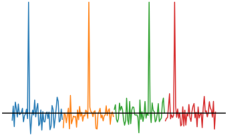
- $I(Z; \langle v, x \rangle) \lesssim b + 1$
  - True but too weak: would get $\Omega(\sqrt{k \log n})$ not $k\sqrt{\log n}$.

What lemma do we want for general k?



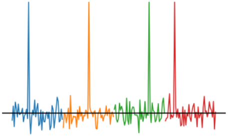
- $I(Z; \langle v, x \rangle) \lesssim b + 1$
  - True but too weak: would get $\Omega(\sqrt{k \log n})$ not $k\sqrt{\log n}$.
- $I(Z; \langle v, x \rangle) \lesssim b/k + 1$

What lemma do we want for general k?



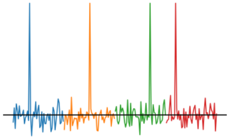
- $I(Z; \langle v, x \rangle) \lesssim b + 1$
  - True but too weak: would get $\Omega(\sqrt{k \log n})$ not $k\sqrt{\log n}$.
- $I(Z; \langle v, x \rangle) \lesssim b/k + 1$
  - Strong but false: if algorithm does 1-sparse recovery on first block, it really can learn $\Theta(b + 1)$ bits.

What lemma do we want for general k?



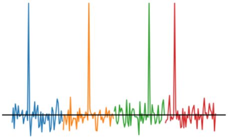
- $I(Z; \langle v, x \rangle) \lesssim b + 1$
  - True but too weak: would get $\Omega(\sqrt{k \log n})$ not $k\sqrt{\log n}$.
- $I(Z; \langle v, x \rangle) \lesssim b/k + 1$
  - Strong but false: if algorithm does 1-sparse recovery on first block, it really can learn $\Theta(b + 1)$ bits.
  - But the learned bits are only about that first block.

What lemma do we want for general k?



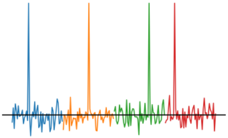
- $I(Z; \langle v, x \rangle) \lesssim b + 1$
  - True but too weak: would get $\Omega(\sqrt{k \log n})$ not $k\sqrt{\log n}$.
- $I(Z; \langle v, x \rangle) \lesssim b/k + 1$
  - Strong but false: if algorithm does 1-sparse recovery on first block, it really can learn $\Theta(b + 1)$ bits.
  - But the learned bits are only about that first block.
- $I(Z_W; \langle v, x \rangle) \lesssim b/k + 1$ for $|W| > 0.99k$.

What lemma do we want for general k?



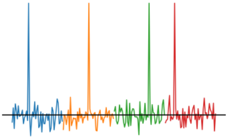
- $I(Z; \langle v, x \rangle) \lesssim b + 1$
  - True but too weak: would get $\Omega(\sqrt{k \log n})$ not $k\sqrt{\log n}$.
- $I(Z; \langle v, x \rangle) \lesssim b/k + 1$
  - Strong but false: if algorithm does 1-sparse recovery on first block, it really can learn $\Theta(b+1)$ bits.
  - But the learned bits are only about that first block.
- $I(Z_W; \langle v, x \rangle) \lesssim b/k + 1$ for $|W| > 0.99k$.
  - Strong enough, at least for constant R.

What lemma do we want for general k?



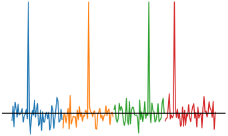
- $I(Z; \langle v, x \rangle) \lesssim b + 1$
  - True but too weak: would get $\Omega(\sqrt{k \log n})$ not $k\sqrt{\log n}$.
- $I(Z; \langle v, x \rangle) \lesssim b/k + 1$
  - Strong but false: if algorithm does 1-sparse recovery on first block, it really can learn $\Theta(b + 1)$ bits.
  - But the learned bits are only about that first block.
- $I(Z_W; \langle v, x \rangle) \lesssim b/k + 1$ for $|W| > 0.99k$.
  - Strong enough, at least for constant R.
  - True for product distributions p...

What lemma do we want for general k?



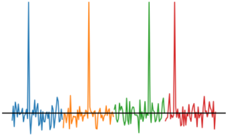
- $I(Z; \langle v, x \rangle) \lesssim b + 1$
  - True but too weak: would get $\Omega(\sqrt{k \log n})$ not $k\sqrt{\log n}$.
- $I(Z; \langle v, x \rangle) \lesssim b/k + 1$
  - Strong but false: if algorithm does 1-sparse recovery on first block, it really can learn $\Theta(b+1)$ bits.
  - But the learned bits are only about that first block.
- $I(Z_W; \langle v, x \rangle) \lesssim b/k + 1$ for $|W| > 0.99k$.
  - Strong enough, at least for constant R.
  - True for product distributions p...
  - but correlated p can make this false.

What lemma do we want for general k?



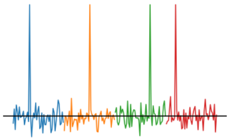
- $I(Z; \langle v, x \rangle) \lesssim b + 1$
  - True but too weak: would get $\Omega(\sqrt{k \log n})$ not $k\sqrt{\log n}$.
- $I(Z; \langle v, x \rangle) \lesssim b/k + 1$
  - Strong but false: if algorithm does 1-sparse recovery on first block, it really can learn $\Theta(b+1)$ bits.
  - But the learned bits are only about that first block.
- $I(Z_W; \langle v, x \rangle) \lesssim b/k + 1$ for $|W| > 0.99k$.
  - Strong enough, at least for constant R.
  - True for product distributions p...
  - but correlated p can make this false.
- $I(Z_W; \langle v, x \rangle) \lesssim b/k + \log k$

What lemma do we want for general k?



- $I(Z; \langle v, x\rangle) \lesssim b + 1$
  - True but too weak: would get $\Omega(\sqrt{k \log n})$ not $k\sqrt{\log n}$.
- $I(Z; \langle v, x\rangle) \lesssim b/k + 1$
  - Strong but false: if algorithm does 1-sparse recovery on first block, it really can learn $\Theta(b+1)$ bits.
  - But the learned bits are only about that first block.
- $I(Z_W; \langle v, x\rangle) \lesssim b/k + 1$ for $|W| > 0.99k$.
  - Strong enough, at least for constant R.
  - True for product distributions p...
  - but correlated p can make this false.
- $I(Z_W; \langle v, x\rangle) \lesssim b/k + \log k$
  - True!

What lemma do we want for general k?



- $I(Z; \langle v, x \rangle) \lesssim b + 1$
  - True but too weak: would get $\Omega(\sqrt{k \log n})$ not $k\sqrt{\log n}$.
- $I(Z; \langle v, x \rangle) \lesssim b/k + 1$
  - Strong but false: if algorithm does 1-sparse recovery on first block, it really can learn $\Theta(b+1)$ bits.
  - But the learned bits are only about that first block.
- $I(Z_W; \langle v, x \rangle) \lesssim b/k + 1$ for $|W| > 0.99k$.
  - Strong enough, at least for constant R.
  - True for product distributions p...
  - but correlated p can make this false.
- $I(Z_W; \langle v, x \rangle) \lesssim b/k + \log k$
  - True!
  - Strong enough if $b > k \log k$ after the first round.

Approach

$$I(Z_W; \langle v, x \rangle) \lesssim b/k + \log k.$$

Approach

$$I(Z_W; \langle v, x \rangle) \lesssim b/k + \log k.$$

- Data processing and Shannon-Hartley:

$$\begin{aligned} I(Z_W; \langle v, x \rangle) &\leqslant I(\sum_{i \in W} v_{Z_i}; (\sum_{i \in W} v_{Z_i}) + \langle v, w \rangle) \\ &\leqslant \frac{1}{2} \log(1 + SNR) \end{aligned}$$

Approach

$$I(Z_W; \langle v, x\rangle) \lesssim b/k + \log k.$$

- Data processing and Shannon-Hartley:

$$I(Z_W; \langle v, x\rangle) \leqslant I(\sum_{i\in W} v_{Z_i}; (\sum_{i\in W} v_{Z_i}) + \langle v, w\rangle)$$
$$\leqslant \frac{1}{2}\log(1 + SNR)$$

where

$$SNR := \frac{\mathbb{E}_{Z\sim p}[(\sum_{i\in W} v_{Z_i})^2]}{\|v\|_2^2/n} \leqslant k\frac{\mathbb{E}_{Z\sim p}[\sum_{i\in W} v_{Z_i}^2]}{\|v\|_2^2/n}$$

Approach

$$I(Z_W; \langle v, x\rangle) \lesssim b/k + \log k.$$

- Data processing and Shannon-Hartley:

$$\begin{aligned} I(Z_W; \langle v, x\rangle) &\leqslant I(\sum_{i\in W} v_{Z_i}; (\sum_{i\in W} v_{Z_i}) + \langle v, w\rangle) \\ &\leqslant \frac{1}{2}\log(1 + SNR) \end{aligned}$$

where

$$\begin{aligned} SNR := \frac{\mathbb{E}_{Z\sim p}[(\sum_{i\in W} v_{Z_i})^2]}{\|v\|_2^2/n} &\leqslant k\frac{\mathbb{E}_{Z\sim p}[\sum_{i\in W} v_{Z_i}^2]}{\|v\|_2^2/n} \\ &\leqslant k\max_{i\in W} SNR(i). \end{aligned}$$

Approach

$$I(Z_W; \langle v, x \rangle) \lesssim b/k + \log k.$$

- Data processing and Shannon-Hartley:

$$I(Z_W; \langle v, x \rangle) \leqslant I(\sum_{i \in W} v_{Z_i}; (\sum_{i \in W} v_{Z_i}) + \langle v, w \rangle)$$
$$\leqslant \frac{1}{2} \log(1 + SNR)$$

where

$$SNR := \frac{\mathbb{E}_{Z \sim p}[(\sum_{i \in W} v_{Z_i})^2]}{\|v\|_2^2/n} \leqslant k \frac{\mathbb{E}_{Z \sim p}[\sum_{i \in W} v_{Z_i}^2]}{\|v\|_2^2/n}$$
$$\leqslant k \max_{i \in W} SNR(i).$$

- So we just need

$$\max_{i \in W} \log(1 + SNR(i)) \lesssim b/k.$$

Approach

- Would like to find a set W such that:

$$\max_{i \in W} \log(1 + SNR(i)) \lesssim b/k.$$

Approach

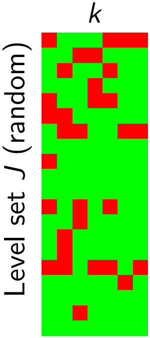
- Would like to find a set W such that:

$$\max_{i \in W} \log(1 + SNR(i)) \lesssim b/k.$$

- What's actually true:

$$\mathop{\mathbb{E}}_{i} \mathop{\mathbb{E}}_{J}[\log(1 + (SNR(i) \mid J))] \lesssim b/k$$

Approach



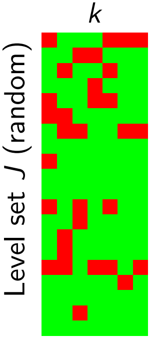
- Would like to find a set W such that:

$$\max_{i \in W} \log(1 + SNR(i)) \lesssim b/k.$$

- What's actually true:

$$\mathop{\mathbb{E}}_{i} \mathop{\mathbb{E}}_{J}[\log(1 + (SNR(i) \mid J))] \lesssim b/k$$

Approach



- Would like to find a set W such that:

$$\max_{i \in W} \log(1 + SNR(i)) \lesssim b/k.$$

- What's actually true:

$$\mathop{\mathbb{E}}_{i} \mathop{\mathbb{E}}_{J}[\log(1 + (SNR(i) \mid J))] \lesssim b/k$$

- Find $W = W(J)$ so that

$$\max_{i \in W} \mathop{\mathbb{E}}_{J} \log(1 + (SNR(i)|J)) \lesssim b/k$$

and $|W| \geqslant 0.99k$ with 99% probability.

Goal for general k

Lemma (Key lemma for general k)

One can choose a set $W = W(J) \subset [k]$ of expected size $0.99k$ so that

$$I(Z_W; Ax) \lesssim m(b/k + \log k) + (b + k)$$

for any $A \in \mathbb{R}^{m \times N}$.

Goal for general k

Lemma (Key lemma for general k)

One can choose a set $W = W(J) \subset [k]$ of expected size $0.99k$ so that

$$I(Z_W; Ax) \lesssim m(b/k + \log k) + (b + k)$$

for any $A \in \mathbb{R}^{m \times N}$.

- Recall $k = 1$ approach:

$$\begin{aligned} I(Z; Ax) &= I(Z; Ax \mid J) + H(J) \\ &\leqslant m \cdot \mathop{\mathbb{E}}_{J}\left[\frac{1}{2}\log(1 + (SNR \mid J))\right] + O(b+1) \\ &\lesssim m(b+1) + (b+1) \end{aligned}$$

Goal for general k

$$I(Z_W; Ax) \lesssim m(b/k + \log k) + (b + k)$$

$k = 1$	General k
$I(Z; Ax)$ $= I(Z; Ax \mid J) + H(J)$ $\lesssim m \cdot \mathbb{E}_J [\log(SNR \mid J)]$ $+ b + 1$ $\lesssim m(b+1) + (b+1)$	

Goal for general k

$$I(Z_W; Ax) \lesssim m(b/k + \log k) + (b + k)$$

$k = 1$	General k
$I(Z; Ax)$	$I(Z_W; Ax)$
$= I(Z; Ax \mid J) + H(J)$	
$\lesssim m \cdot \mathbb{E}_J[\log(SNR \mid J)]$	
$+ b + 1$	
$\lesssim m(b+1) + (b+1)$	

Goal for general k

$$I(Z_W; Ax) \lesssim m(b/k + \log k) + (b + k)$$

$k = 1$	General k
$I(Z; Ax)$	$I(Z_W; Ax)$
$= I(Z; Ax \mid J) + H(J)$	$= I(Z_W; Ax \mid J) + H(J)$
$\lesssim m \cdot \mathbb{E}_J[\log(SNR \mid J)]$	
$+ b + 1$	
$\lesssim m(b+1) + (b+1)$	

Goal for general k

$$I(Z_W; Ax) \lesssim m(b/k + \log k) + (b + k)$$

$k = 1$	General k
$I(Z; Ax)$	$I(Z_W; Ax)$
$= I(Z; Ax \mid J) + H(J)$	$= I(Z_W; Ax \mid J) + H(J)$
$\lesssim m \cdot \mathbb{E}_J [\log(SNR \mid J)]$	$\lesssim m \cdot \mathbb{E}_J \left[\log(SNR(\bigcup_{i \in W} Z_i) \mid J)\right]$
$+ b + 1$	$+ b + k$
$\lesssim m(b+1) + (b+1)$	

Goal for general k

$$I(Z_W; Ax) \lesssim m(b/k + \log k) + (b + k)$$

$k = 1$	General k
$I(Z; Ax)$	$I(Z_W; Ax)$
$= I(Z; Ax \mid J) + H(J)$	$= I(Z_W; Ax \mid J) + H(J)$
$\lesssim m \cdot \mathbb{E}_J [\log(SNR \mid J)]$	$\lesssim m \cdot \mathbb{E}_J \left[\log(SNR(\bigcup_{i \in W} Z_i) \mid J)\right]$
$+ b + 1$	$+ b + k$
	$\lesssim m \cdot \mathbb{E}_J \left[\log(k \cdot \max_{i \in W} SNR(Z_i) \mid J)\right]$
$\lesssim m(b + 1) + (b + 1)$	

Goal for general k

$$I(Z_W; Ax) \lesssim m(b/k + \log k) + (b + k)$$

$k = 1$	General k
$I(Z; Ax)$	$I(Z_W; Ax)$
$= I(Z; Ax \mid J) + H(J)$	$= I(Z_W; Ax \mid J) + H(J)$
$\lesssim m \cdot \mathbb{E}_J [\log(SNR \mid J)]$	$\lesssim m \cdot \mathbb{E}_J \left[\log(SNR(\bigcup_{i \in W} Z_i) \mid J)\right]$
$+ b + 1$	$+ b + k$
	$\lesssim m \cdot \mathbb{E}_J \left[\log(k \cdot \max_{i \in W} SNR(Z_i) \mid J)\right]$
$\lesssim m(b + 1) + (b + 1)$	$\lesssim m \cdot (b/k + \log k) + (b + k)$

Wrapping up the lower bound for $R = 2$

- Suppose $m > k \log k$ measurements per round.

Wrapping up the lower bound for $R = 2$

- Suppose $m > k \log k$ measurements per round.
- First round is nonadaptive: learn $b = O(m)$ bits.

Wrapping up the lower bound for $R = 2$

- Suppose $m > k \log k$ measurements per round.
- First round is nonadaptive: learn $b = O(m)$ bits.
- Second round, learn

$$m(b/k + \log k) + (b + k) = O(m^2/k)$$

bits.

Wrapping up the lower bound for $R = 2$

- Suppose $m > k \log k$ measurements per round.
- First round is nonadaptive: learn $b = O(m)$ bits.
- Second round, learn

$$m(b/k + \log k) + (b + k) = O(m^2/k)$$

  bits.
- But need to learn $|W| \log n \approx k \log n$ bits.

Wrapping up the lower bound for $R = 2$

- Suppose $m > k \log k$ measurements per round.
- First round is nonadaptive: learn $b = O(m)$ bits.
- Second round, learn

  $$m(b/k + \log k) + (b + k) = O(m^2/k)$$

  bits.
- But need to learn $|W| \log n \approx k \log n$ bits.
- Hence $m \gtrsim k\sqrt{\log n}$ (if this is more than $k \log k$).

Wrapping up the lower bound for $R = 2$

- Suppose $m > k \log k$ measurements per round.
- First round is nonadaptive: learn $b = O(m)$ bits.
- Second round, learn

$$m(b/k + \log k) + (b + k) = O(m^2/k)$$

  bits.
- But need to learn $|W| \log n \approx k \log n$ bits.
- Hence $m \gtrsim k\sqrt{\log n}$ (if this is more than $k \log k$).
- Open questions:

Wrapping up the lower bound for $R = 2$

- Suppose $m > k \log k$ measurements per round.
- First round is nonadaptive: learn $b = O(m)$ bits.
- Second round, learn

  $$m(b/k + \log k) + (b + k) = O(m^2/k)$$

  bits.
- But need to learn $|W| \log n \approx k \log n$ bits.
- Hence $m \gtrsim k\sqrt{\log n}$ (if this is more than $k \log k$).
- Open questions:
  - Less restriction on k? Conjecture:

    $$I(Z_W; Ax \mid Z_{\overline{W}}) \lesssim b/k + 1$$

Wrapping up the lower bound for $R = 2$

- Suppose $m > k \log k$ measurements per round.
- First round is nonadaptive: learn $b = O(m)$ bits.
- Second round, learn

  $$m(b/k + \log k) + (b + k) = O(m^2/k)$$

  bits.
- But need to learn $|W| \log n \approx k \log n$ bits.
- Hence $m \gtrsim k\sqrt{\log n}$ (if this is more than $k \log k$).
- Open questions:
  - Less restriction on k? Conjecture:

    $$I(Z_W; Ax \mid Z_{\overline{W}}) \lesssim b/k + 1$$

  - Better dependence on R?

Outline

Standard sparse recovery approach

- We have an optimal adaptive 1-sparse recovery algorithm.

Standard sparse recovery approach

- We have an optimal adaptive 1-sparse recovery algorithm.
- Standard technique:
  1. Throw coordinates into buckets.
  2. 1-sparse recovery within each bucket.
  3. Clean up mistakes.

Standard sparse recovery approach

- We have an optimal adaptive 1-sparse recovery algorithm.
- Standard technique:
  1. Throw coordinates into buckets.
  2. 1-sparse recovery within each bucket.
  3. Clean up mistakes.
- Problem: surrounding steps add rounds.
  - [IPW '11]: cleanup is recursive, multiplying rounds by $O(\log^* k)$.
  - [NSZW '18]: 1 round setup, 2 rounds cleanup.

Standard sparse recovery approach

- We have an optimal adaptive 1-sparse recovery algorithm.
- Standard technique:
  1. Throw coordinates into buckets.
  2. 1-sparse recovery within each bucket.
  3. Clean up mistakes.
- Problem: surrounding steps add rounds.
  - [IPW '11]: cleanup is recursive, multiplying rounds by $O(\log^* k)$.
  - [NSZW '18]: 1 round setup, 2 rounds cleanup.
- Our approach: avoid reduction to $k = 1$.

Standard sparse recovery approach

- We have an optimal adaptive 1-sparse recovery algorithm.
- Standard technique:
  1. Throw coordinates into buckets.
  2. 1-sparse recovery within each bucket.
  3. Clean up mistakes.
- Problem: surrounding steps add rounds.
  - [IPW '11]: cleanup is recursive, multiplying rounds by $O(\log^* k)$.
  - [NSZW '18]: 1 round setup, 2 rounds cleanup.
- Our approach: avoid reduction to $k = 1$.
  - Instead, reduce to C-approximate k-sparse recovery for $C \gg 1$.

Standard sparse recovery approach

- We have an optimal adaptive 1-sparse recovery algorithm.
- Standard technique:
  1. Throw coordinates into buckets.
  2. 1-sparse recovery within each bucket.
  3. Clean up mistakes.
- Problem: surrounding steps add rounds.
  - [IPW '11]: cleanup is recursive, multiplying rounds by $O(\log^* k)$.
  - [NSZW '18]: 1 round setup, 2 rounds cleanup.
- Our approach: avoid reduction to $k = 1$.
  - Instead, reduce to C-approximate k-sparse recovery for $C \gg 1$.
  - This is solvable nonadaptively in $O(k \log_C(n/k) \cdot \log^* k)$ measurements. [Price-Woodruff '12]

Basic approach: $R = 2$

1. Throw the coordinates into $B = k \cdot 2^{\sqrt{\log n}}$ buckets, and nonadaptively apply k-sparse $O(1)$-approximate recovery.
   - $k \log(B/k) = k\sqrt{\log n}$ measurements.

Basic approach: $R = 2$

1. Throw the coordinates into $B = k \cdot 2^{\sqrt{\log n}}$ buckets, and nonadaptively apply k-sparse $O(1)$-approximate recovery.
   - $k \log(B/k) = k\sqrt{\log n}$ measurements.
2. Apply k-sparse $2^{\sqrt{\log n}}$-approximate recovery to the preimage.
   - $k \log_C n = k\sqrt{\log n}$ measurements.

Basic approach: $R = 2$

1. Throw the coordinates into $B = k \cdot 2^{\sqrt{\log n}}$ buckets, and nonadaptively apply k-sparse $O(1)$-approximate recovery.
   - $k \log(B/k) = k\sqrt{\log n}$ measurements.
2. Apply k-sparse $2^{\sqrt{\log n}}$-approximate recovery to the preimage.
   - $k \log_C n = k\sqrt{\log n}$ measurements.

- Key problem: can't miss anything important in the first round.

Basic approach: $R = 2$

1. Throw the coordinates into $B = k \cdot 2^{\sqrt{\log n}}$ buckets, and nonadaptively apply k-sparse $O(1)$-approximate recovery.
   - $k \log(B/k) = k\sqrt{\log n}$ measurements.
2. Apply k-sparse $2^{\sqrt{\log n}}$-approximate recovery to the preimage.
   - $k \log_C n = k\sqrt{\log n}$ measurements.

- Key problem: can't miss anything important in the first round.
  - There *will* be collisions.

Basic approach: $R = 2$

1. Throw the coordinates into $B = k \cdot 2^{\sqrt{\log n}}$ buckets, and nonadaptively apply k-sparse $O(1)$-approximate recovery.
   - $k \log(B/k) = k\sqrt{\log n}$ measurements.
2. Apply k-sparse $2^{\sqrt{\log n}}$-approximate recovery to the preimage.
   - $k \log_C n = k\sqrt{\log n}$ measurements.

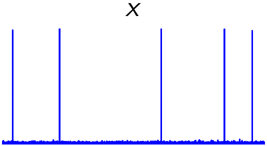
- Key problem: can't miss anything important in the first round.
  - There *will* be collisions.
  - Yet if x has no noise, must find every entry.

Basic approach: $R = 2$

1. Throw the coordinates into $B = k \cdot 2^{\sqrt{\log n}}$ buckets, and nonadaptively apply k-sparse $O(1)$-approximate recovery.
   - $k \log(B/k) = k\sqrt{\log n}$ measurements.
2. Apply k-sparse $2^{\sqrt{\log n}}$-approximate recovery to the preimage.
   - $k \log_C n = k\sqrt{\log n}$ measurements.

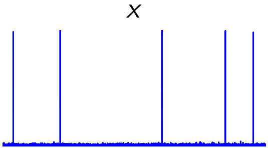
- Key problem: can't miss anything important in the first round.
  - There *will* be collisions.
  - Yet if x has no noise, must find every entry.
- Solution: *triple Gaussian hashing*.

Hashing



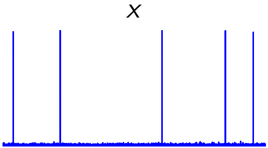
- For intuition, consider x being k-sparse binary + Gaussian with norm 1.

Hashing



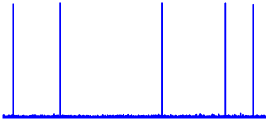
- For intuition, consider x being k-sparse binary $+$ Gaussian with norm 1.
  - Successful recovery must find all but $O(1)$ binary entries of x.

Hashing



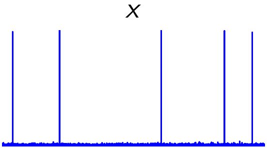
- For intuition, consider x being k-sparse binary + Gaussian with norm 1.
  - Successful recovery must find all but $O(1)$ binary entries of x.
- Given partition $h : [n] \to [B]$, how to condense $x \in \mathbb{R}^n$ into $y \in \mathbb{R}^B$?

Hashing



- For intuition, consider x being k-sparse binary + Gaussian with norm 1.
  - Successful recovery must find all but $O(1)$ binary entries of x.
- Given partition $h : [n] \to [B]$, how to condense $x \in \mathbb{R}^n$ into $y \in \mathbb{R}^B$?
  - Goal: preimage of k-sparse recovery on y includes large entries in x.

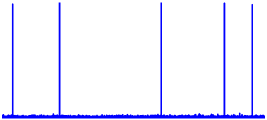
Hashing



- For intuition, consider x being k-sparse binary + Gaussian with norm 1.
  - Successful recovery must find all but $O(1)$ binary entries of x.
- Given partition $h : [n] \to [B]$, how to condense $x \in \mathbb{R}^n$ into $y \in \mathbb{R}^B$?
  - Goal: preimage of k-sparse recovery on y includes large entries in x.
- Random signs: $s : [n] \to \{\pm 1\}$ and

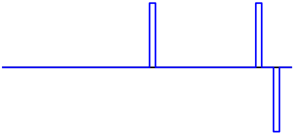
$$y_u = \sum_{i:h(i)=u} x_i \cdot s(i).$$

Hashing



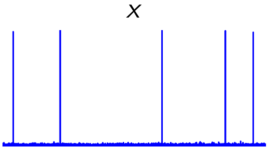
- For intuition, consider x being k-sparse binary + Gaussian with norm 1.
  - Successful recovery must find all but $O(1)$ binary entries of x.
- Given partition $h : [n] \to [B]$, how to condense $x \in \mathbb{R}^n$ into $y \in \mathbb{R}^B$?
  - Goal: preimage of k-sparse recovery on y includes large entries in x.
- Random signs: $s : [n] \to \{\pm 1\}$ and

$$y_u = \sum_{i:h(i)=u} x_i \cdot s(i).$$



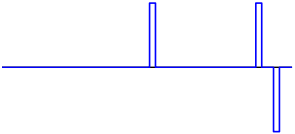
Without noise

Hashing

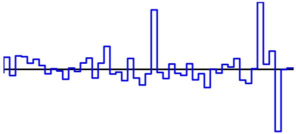


- For intuition, consider x being k-sparse binary + Gaussian with norm 1.
  - Successful recovery must find all but $O(1)$ binary entries of x.
- Given partition $h : [n] \to [B]$, how to condense $x \in \mathbb{R}^n$ into $y \in \mathbb{R}^B$?
  - Goal: preimage of k-sparse recovery on y includes large entries in x.
- Random signs: $s : [n] \to \{\pm 1\}$ and

$$y_u = \sum_{i : h(i) = u} x_i \cdot s(i).$$

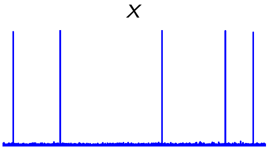


Without noise



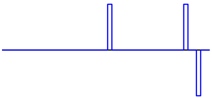
With noise

Gaussian hashing

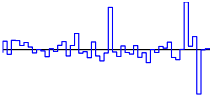


- Random signs: $s : [n] \to \{\pm 1\}$ and

$$y_u = \sum_{i:h(i)=u} x_i \cdot s(i).$$

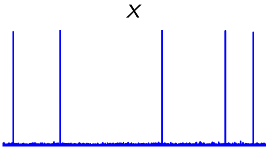


Without noise



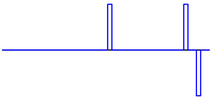
With noise

Gaussian hashing

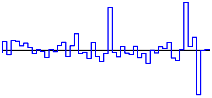


- Random signs: $s : [n] \to \{\pm 1\}$ and

$$y_u = \sum_{i : h(i) = u} x_i \cdot s(i).$$



Without noise

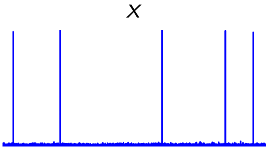


With noise

- Gaussian hashing: $g \sim N(I_n)$ and

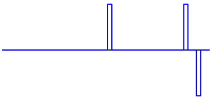
$$y_u = \sum_{i : h(i) = u} x_i \cdot g(i).$$

Gaussian hashing

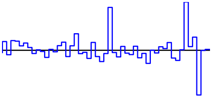


- Random signs: $s : [n] \to \{\pm 1\}$ and

$$y_u = \sum_{i:h(i)=u} x_i \cdot s(i).$$



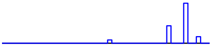
Without noise



With noise

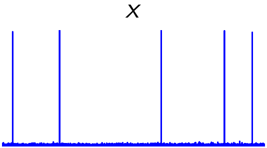
- Gaussian hashing: $g \sim N(I_n)$ and

$$y_u = \sum_{i:h(i)=u} x_i \cdot g(i).$$



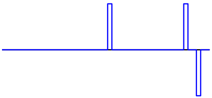
Without noise

Gaussian hashing

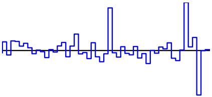


- Random signs: $s : [n] \to \{\pm 1\}$ and

$$y_u = \sum_{i:h(i)=u} x_i \cdot s(i).$$



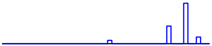
Without noise



With noise

- Gaussian hashing: $g \sim N(I_n)$ and

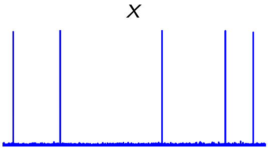
$$y_u = \sum_{i:h(i)=u} x_i \cdot g(i).$$



Without noise

With noise

Triple Gaussian hashing

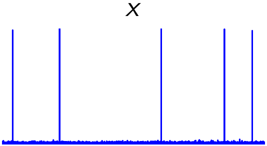


- *Triple* Gaussian hashing: $g^1, g^2, g^3 \sim N(0, I_n)$;

$$y^j_u = \sum_{i:h(i)=u} x_i \cdot g^j(i).$$

Try 1

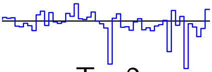
Triple Gaussian hashing



- *Triple* Gaussian hashing: $g^1, g^2, g^3 \sim N(0, I_n)$;

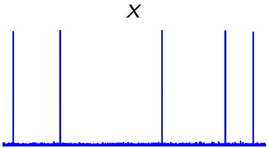
$$y_u^j = \sum_{i:h(i)=u} x_i \cdot g^j(i).$$

Try 1



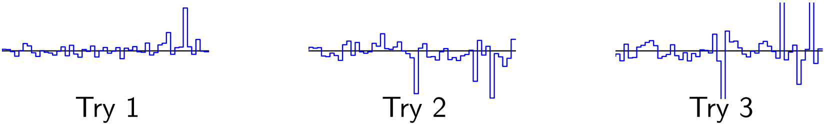
Try 2

Triple Gaussian hashing

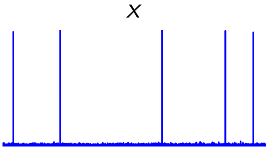


- *Triple* Gaussian hashing: $g^1, g^2, g^3 \sim N(0, I_n)$;

$$y_u^j = \sum_{i:h(i)=u} x_i \cdot g^j(i).$$

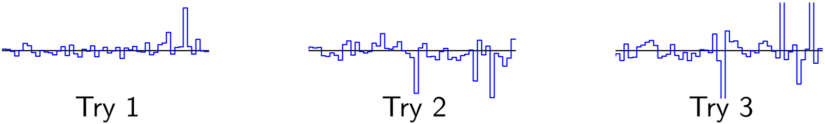


Triple Gaussian hashing



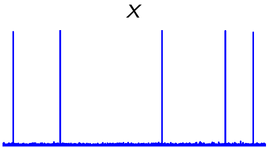
- *Triple* Gaussian hashing: $g^1, g^2, g^3 \sim N(0, I_n)$;

$$y_u^j = \sum_{i:h(i)=u} x_i \cdot g^j(i).$$



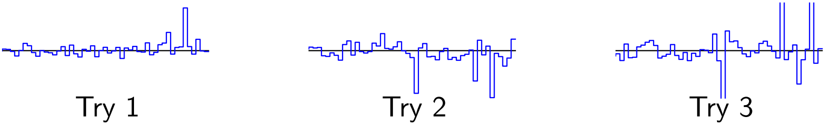
  - Take union of three independent sparse recovery attempts.

Triple Gaussian hashing



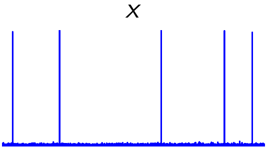
- *Triple* Gaussian hashing: $g^1, g^2, g^3 \sim N(0, I_n)$;

$$y_u^j = \sum_{i:h(i)=u} x_i \cdot g^j(i).$$



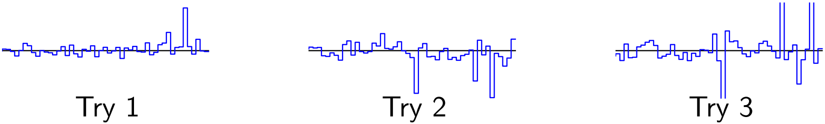
  - Take union of three independent sparse recovery attempts.
  - Expected false negatives are $O(\text{noise})$, so can be skipped.

Triple Gaussian hashing



- *Triple* Gaussian hashing: $g^1, g^2, g^3 \sim N(0, I_n)$;

$$y_u^j = \sum_{i:h(i)=u} x_i \cdot g^j(i).$$



  - Take union of three independent sparse recovery attempts.
  - Expected false negatives are $O(\text{noise})$, so can be skipped.

- Avoids the cleanup rounds, getting

$$O(k \log^{1/R} n \cdot \log^* k)$$

measurements.

Results

- Previously:

$$k + \log^{1/R} n \lesssim m \lesssim k \cdot \log^{1/(R-3)} n$$

Results

- Previously:

$$k + \log^{1/R} n \lesssim m \lesssim k \cdot \log^{1/(R-3)} n$$

- Now:

$$k \cdot \log^{1/R} n \lesssim m \lesssim k \cdot \log^{1/R} n \cdot \log^* k$$

where the lower bound applies if this is above $k \log k$.

Results

- Previously:
$$k + \log^{1/R} n \lesssim m \lesssim k \cdot \log^{1/(R-3)} n$$
- Now:
$$k \cdot \log^{1/R} n \lesssim m \lesssim k \cdot \log^{1/R} n \cdot \log^* k$$
where the lower bound applies if this is above $k \log k$.
- Biggest question:
  - Are $\omega(k)$ measurements necessary for unlimited R?

Results

- Previously:

$$k + \log^{1/R} n \lesssim m \lesssim k \cdot \log^{1/(R-3)} n$$

- Now:

$$k \cdot \log^{1/R} n \lesssim m \lesssim k \cdot \log^{1/R} n \cdot \log^* k$$

  where the lower bound applies if this is above $k \log k$.
- Biggest question:
  - Are $\omega(k)$ measurements necessary for unlimited R?

Thank You