Sparse recovery with partial support knowledge

Khanh Do Ba Piotr Indyk

August 17, 2011

Sparse recovery

Measurement:

- Data/signal in n-dimensional space: x
- Goal: compress x into a "sketch" Ax, where A is an $m \times n$ matrix, $m \ll n$

$$\begin{pmatrix} & A & \end{pmatrix} \begin{pmatrix} \\ x \\ \\ \end{pmatrix} = \begin{pmatrix} Ax \end{pmatrix}$$

Sparse recovery

Measurement:

- Data/signal in n-dimensional space: x
- Goal: compress x into a "sketch" Ax, where A is an $m \times n$ matrix, $m \ll n$

$$\begin{pmatrix} & A & \end{pmatrix} \begin{pmatrix} \\ x \\ \\ \end{pmatrix} = \begin{pmatrix} Ax \end{pmatrix}$$

Recovery:

- Sparsity parameter k
- Informal: recover largest k coordinates of x
- Formal: recover approximation $\hat{x}$ of x such that

$$\|\hat{x} - x\|_p \leq C(k) \min_{x'} \|x' - x\|_q$$

over all k-sparse x' (at most k non-zero coordinates)

Sparse recovery

Measurement:

- Data/signal in n-dimensional space: x
- Goal: compress x into a "sketch" Ax, where A is an $m \times n$ matrix, $m \ll n$

$$\begin{pmatrix} & A & \end{pmatrix} \begin{pmatrix} \\ x \\ \\ \end{pmatrix} = \begin{pmatrix} Ax \end{pmatrix}$$

Recovery:

- Sparsity parameter k
- Informal: recover largest k coordinates of x
- Formal: recover approximation $\hat{x}$ of x such that

$$\|\hat{x} - x\|_p \leq C(k) \min_{x'} \|x' - x\|_q$$

over all k-sparse x' (at most k non-zero coordinates)

Want:

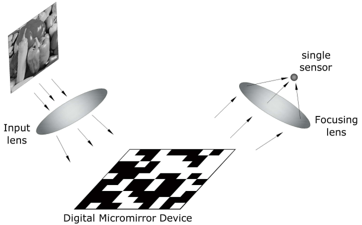
- Good compression: small $m = m(k, n)$
- Efficient algorithms for encoding and recovery

Applications

- Monitoring network traffic data streams
  - x is traffic matrix, for every source/destination pair
  - Too big to store!
  - Need to compress yet allow quick updates
  - Linear compression allows quick update: $A(x + \Delta) = Ax + A\Delta$

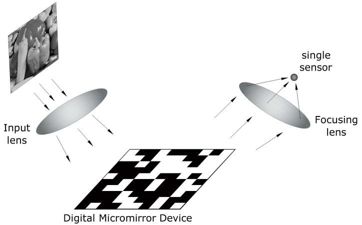
Applications

- Monitoring network traffic data streams
  - x is traffic matrix, for every source/destination pair
  - Too big to store!
  - Need to compress yet allow quick updates
  - Linear compression allows quick update: $A(x + \Delta) = Ax + A\Delta$
- Single-pixel camera (Wakin et al.'06)



Applications

- Monitoring network traffic data streams
  - x is traffic matrix, for every source/destination pair
  - Too big to store!
  - Need to compress yet allow quick updates
  - Linear compression allows quick update: $A(x+\Delta) = Ax + A\Delta$
- Single-pixel camera (Wakin et al.'06)



- Various forms of pooling experiments

Previous results in sparse recovery

Paper	Sketch length	Approx
CRT'04	$O(k \log(n/k))$	$\ell_1 \leq O(1)\ell_1$
GLPS'10	$O((k/\epsilon)\log(n/k))$	$\ell_2 \leq (1+\epsilon)\ell_2$
DIPW'10,FPRU'10	$\Omega(k \log(n/k))$	$\ell_1 \leq O(1)\ell_1$, $\ell_2 \leq O(1)\ell_2$

Previous results in sparse recovery

Paper	Sketch length	Approx
CRT'04	$O(k \log(n/k))$	$\ell_1 \leq O(1)\ell_1$
GLPS'10	$O((k/\epsilon) \log(n/k))$	$\ell_2 \leq (1+\epsilon)\ell_2$
DIPW'10,FPRU'10	$\Omega(k \log(n/k))$	$\ell_1 \leq O(1)\ell_1$, $\ell_2 \leq O(1)\ell_2$

Can we do better than $O(k \log(n/k))$?

Previous results in sparse recovery

Paper	Sketch length	Approx
CRT'04	$O(k \log(n/k))$	$\ell_1 \leq O(1)\ell_1$
GLPS'10	$O((k/\epsilon) \log(n/k))$	$\ell_2 \leq (1+\epsilon)\ell_2$
DIPW'10,FPRU'10	$\Omega(k \log(n/k))$	$\ell_1 \leq O(1)\ell_1$, $\ell_2 \leq O(1)\ell_2$

Can we do better than $O(k \log(n/k))$?
Yes! With additional knowledge about the signal.

Partial knowledge

- Model-based compressive sensing (Baraniuk et al.'10, Eldar-Bolcskei'09)

  sets of large coefficients known to exhibit some patterns

Partial knowledge

- Model-based compressive sensing (Baraniuk et al.'10, Eldar-Bolcskei'09)

  sets of large coefficients known to exhibit some patterns

- Bayesian compressive sensing (Cevher et al.'10)

  signal generated from known distribution

Partial knowledge

- Model-based compressive sensing (Baraniuk et al.'10, Eldar-Bolcskei'09)

    sets of large coefficients known to exhibit some patterns

- Bayesian compressive sensing (Cevher et al.'10)

    signal generated from known distribution

- Support knowledge (Price'10, this paper)

    some knowledge available about where the large coefficients lie

Partial support knowledge

Sparse recovery with partial support knowledge (SRPSK):

1. Construction of A: parameters n, k and s
2. Measurement: Ax
3. Support knowledge: set $S \subset [n]$, $|S| = s$, where top-k "likely" lies
4. Recovery: from Ax and S, find $\hat{x}$ such that

$$\|\hat{x} - x\|_p \leq C \min_{x'} \|x' - x\|_q$$

over all k-sparse x' with support in S

Motivation

Applications:

- Tracking tasks: object position typically does not change quickly
- Exploratory tasks: support lies in one of several sets

Motivation

Applications:

- Tracking tasks: object position typically does not change quickly
- Exploratory tasks: support lies in one of several sets

Theoretical:

- $s = n$: "regular" sparse recovery
- $s = k$: set query (Price'10)

Our results

Theorem (Upper bound)

SRPSK with the ℓ_2/ℓ_2 guarantee can be solved $(1+\epsilon)$-approximately using $O((k/\epsilon)\log(s/k))$ measurements.

Our results

Theorem (Upper bound)

SRPSK with the ℓ_2/ℓ_2 guarantee can be solved $(1+\epsilon)$-approximately using $O((k/\epsilon)\log(s/k))$ measurements.

Theorem (Lower bound)

Any $(1+\epsilon)$-approximate solution to SRPSK with either the ℓ_1/ℓ_1 or the ℓ_2/ℓ_2 guarantee requires $\Omega((k/\epsilon)\log(s/k))$ measurements, assuming $s = O(\epsilon n/\log(n/\epsilon))$.

Proof sketch: upper bound

Noise-tolerant sparse recovery: recover $\hat{x}$ from $Ax + \nu$ such that

$$\|\hat{x} - x\|_p \leq (1 + \epsilon) \min_{x'} \|x' - x\|_p + \epsilon \|\nu\|_p$$

where $\mathbb{E}[\|Av\|_p] \leq \|v\|_p$ for every v.

Proof sketch: upper bound

Noise-tolerant sparse recovery: recover $\hat{x}$ from $Ax + \nu$ such that

$$\|\hat{x} - x\|_p \leq (1 + \epsilon) \min_{x'} \|x' - x\|_p + \epsilon\|\nu\|_p$$

where $\mathbb{E}[\|Av\|_p] \leq \|v\|_p$ for every v.

GLPS'10 gives this guarantee!

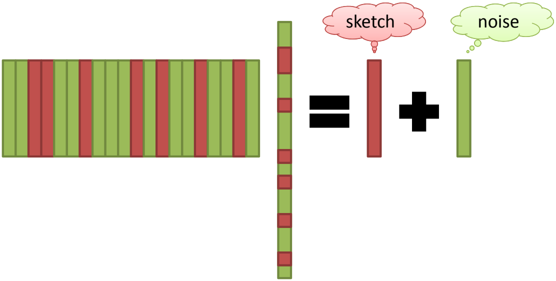
Proof sketch: upper bound

Noise-tolerant sparse recovery: recover $\hat{x}$ from $Ax + \nu$ such that

$$\|\hat{x} - x\|_p \leq (1+\epsilon) \min_{x'} \|x' - x\|_p + \epsilon\|\nu\|_p$$

where $\mathbb{E}[\|Av\|_p] \leq \|v\|_p$ for every v.

GLPS'10 gives this guarantee!



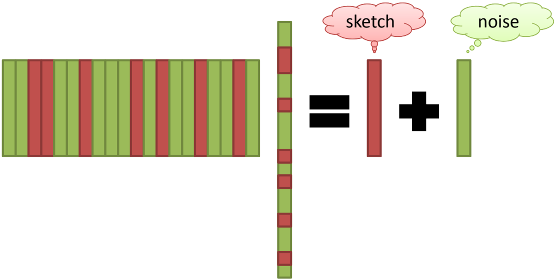
Proof sketch: upper bound

Noise-tolerant sparse recovery: recover $\hat{x}$ from $Ax + \nu$ such that

$$\|\hat{x} - x\|_p \leq (1+\epsilon) \min_{x'} \|x' - x\|_p + \epsilon \|\nu\|_p$$

where $\mathbb{E}[\|Av\|_p] \leq \|v\|_p$ for every v.

GLPS'10 gives this guarantee!



How to construct matrix so that specifying any s columns yields a "good" measurement (sub)matrix?

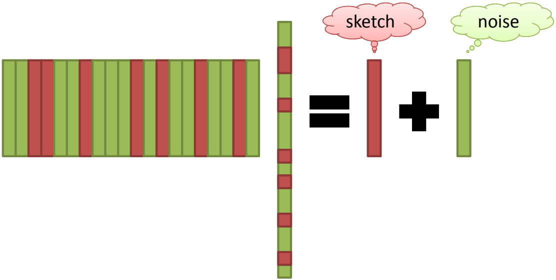
Proof sketch: upper bound

Noise-tolerant sparse recovery: recover $\hat{x}$ from $Ax + \nu$ such that

$$\|\hat{x} - x\|_p \leq (1+\epsilon) \min_{x'} \|x' - x\|_p + \epsilon \|\nu\|_p$$

where $\mathbb{E}[\|Av\|_p] \leq \|v\|_p$ for every v.

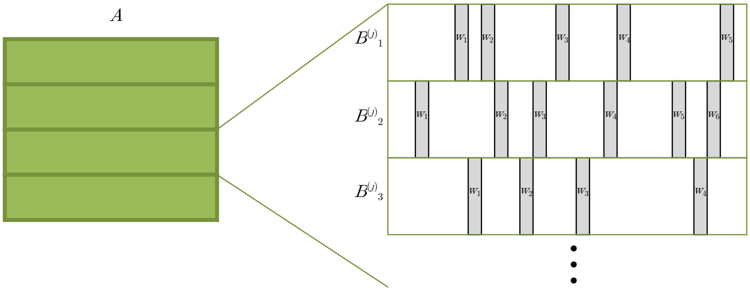
GLPS'10 gives this guarantee!



How to construct matrix so that specifying any s columns yields a "good" measurement (sub)matrix? Independently generated columns!

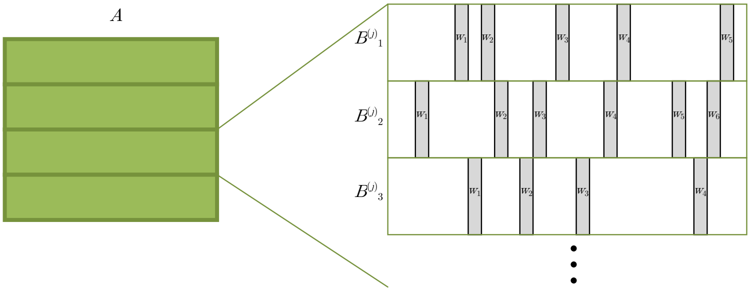
Proof sketch: upper bound (cont.)

Part of GLPS'10 measurement matrix that is not independent:



Proof sketch: upper bound (cont.)

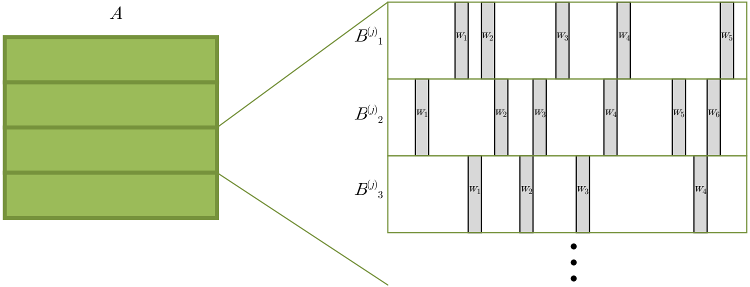
Part of GLPS'10 measurement matrix that is not independent:



- Code words $w_1, w_2, \ldots$ only need to be distinct within each "block"

Proof sketch: upper bound (cont.)

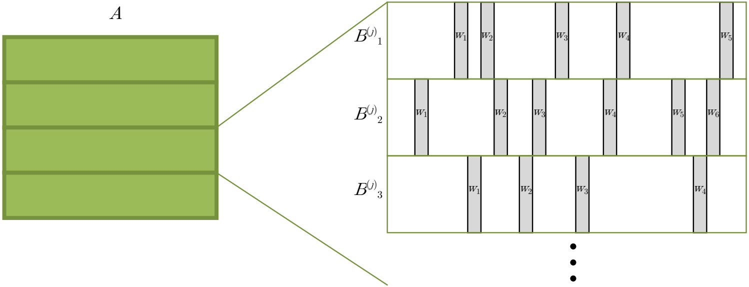
Part of GLPS'10 measurement matrix that is not independent:



- Code words $w_1, w_2, \ldots$ only need to be distinct within each "block"
- If h code words used, pick independently from $O(h^3)$ code words instead

Proof sketch: upper bound (cont.)

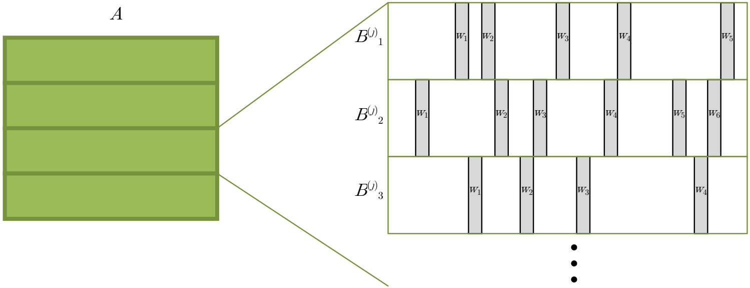
Part of GLPS'10 measurement matrix that is not independent:



- Code words $w_1, w_2, \ldots$ only need to be distinct within each "block"
- If h code words used, pick independently from $O(h^3)$ code words instead
- If b blocks used, construct $2b$ blocks instead and, given S, pick b "good" blocks to use

Proof sketch: upper bound (cont.)

Part of GLPS'10 measurement matrix that is not independent:



- Code words $w_1, w_2, \ldots$ only need to be distinct within each "block"
- If h code words used, pick independently from $O(h^3)$ code words instead
- If b blocks used, construct $2b$ blocks instead and, given S, pick b "good" blocks to use
- Result: about 6 times as many rows, but columns now independent!

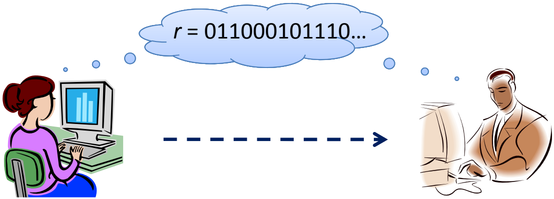
Proof sketch: lower bound

Based on proof in DIPW'10 of lower bound for sparse recovery

Proof sketch: lower bound

Based on proof in DIPW'10 of lower bound for sparse recovery

Augmented Indexing:



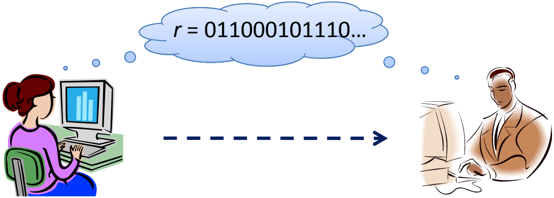
Input: $b = (b_1, b_2, \ldots, b_d) \in \{0,1\}^d$

Input: $i \in [d], b_{i+1}, b_{i+2}, \ldots, b_d$
Output: b_i

Proof sketch: lower bound

Based on proof in DIPW'10 of lower bound for sparse recovery

Augmented Indexing:

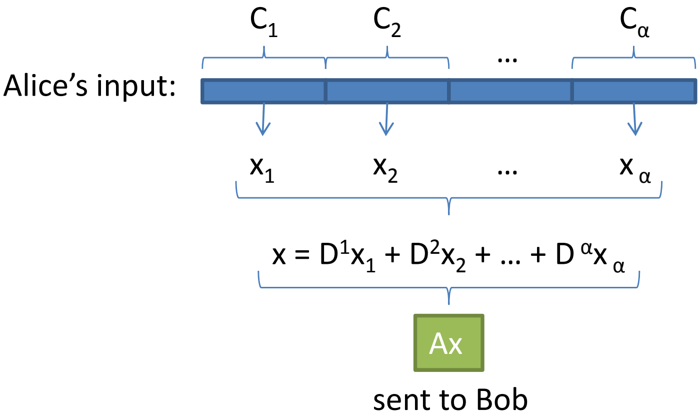


Input: $b = (b_1, b_2, \ldots, b_d) \in \{0,1\}^d$

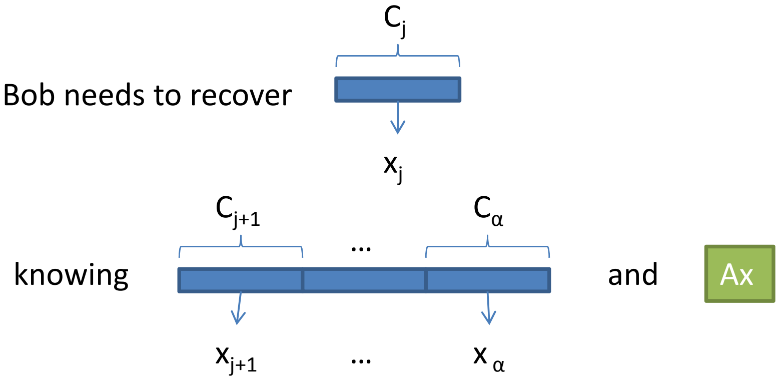
Input: $i \in [d]$, $b_{i+1}, b_{i+2}, \ldots, b_d$
Output: b_i

Known: requires $\Omega(d)$ bits of communication

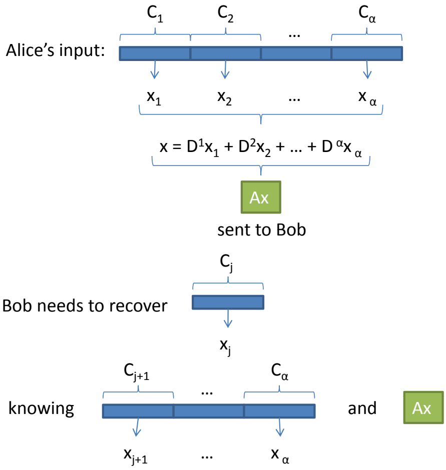
Proof sketch: lower bound (cont.)



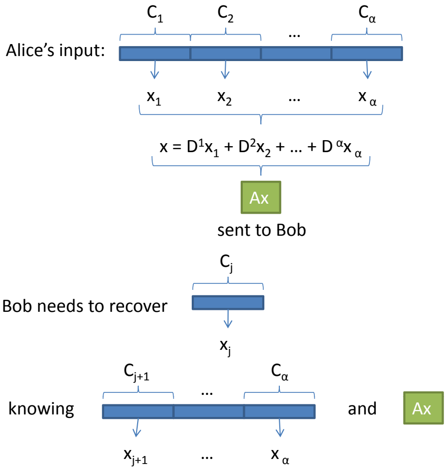
Proof sketch: lower bound (cont.)



Proof sketch: lower bound (cont.)

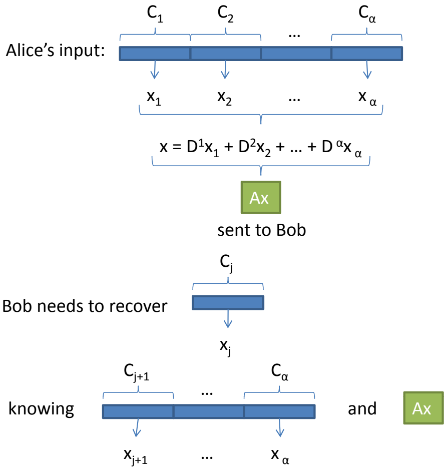


Proof sketch: lower bound (cont.)



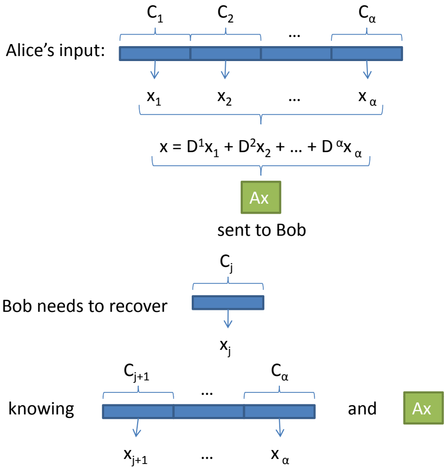
- From linearity Bob can compute $A(D^1 x_1 + \cdots + D^j x_j)$

Proof sketch: lower bound (cont.)



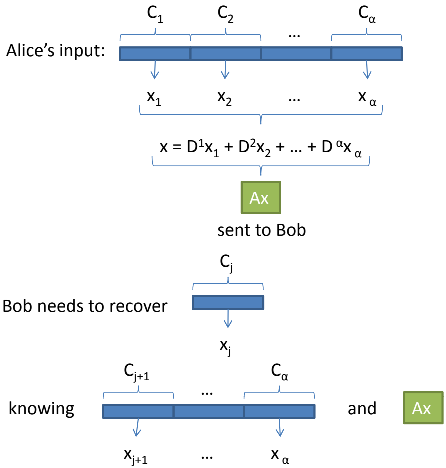
- From linearity Bob can compute $A(D^1x_1 + \cdots + D^jx_j)$
- "Signal" $D^1x_1 + \cdots + D^jx_j$ dominated by D^jx_j with support in j^{th} block S_j

Proof sketch: lower bound (cont.)



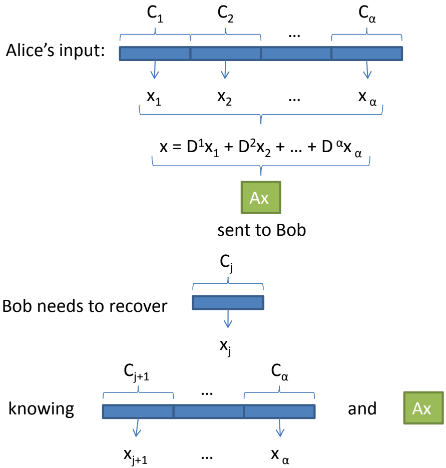
- From linearity Bob can compute $A(D^1x_1 + \cdots + D^jx_j)$
- "Signal" $D^1x_1 + \cdots + D^jx_j$ dominated by D^jx_j with support in j^{th} block S_j
- Use support knowledge to recover approximate x_j

Proof sketch: lower bound (cont.)



- From linearity Bob can compute $A(D^1 x_1 + \cdots + D^j x_j)$
- "Signal" $D^1 x_1 + \cdots + D^j x_j$ dominated by $D^j x_j$ with support in j^{th} block S_j
- Use support knowledge to recover approximate x_j
- Use error correction to recover exact x_j

Proof sketch: lower bound (cont.)



- From linearity Bob can compute $A(D^1 x_1 + \cdots + D^j x_j)$
- "Signal" $D^1 x_1 + \cdots + D^j x_j$ dominated by $D^j x_j$ with support in j^{th} block S_j
- Use support knowledge to recover approximate x_j
- Use error correction to recover exact x_j
- Lower bound of AI implies lower bound for SRPSK