CS on CS: Computer Science insights into Compresive Sensing (and vice versa)

Piotr Indyk

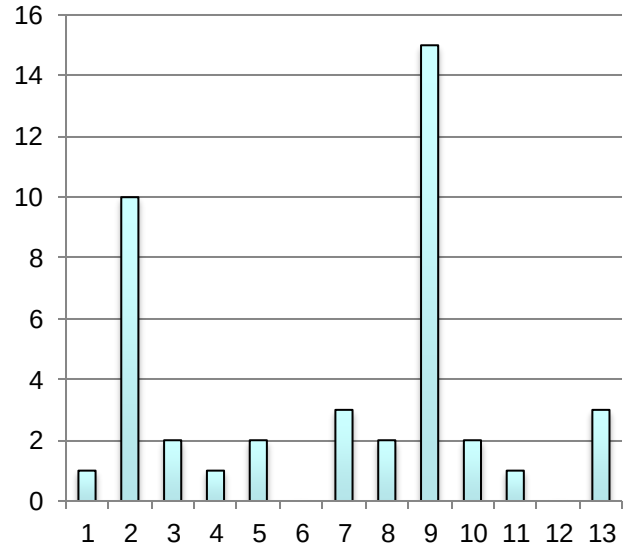
MIT

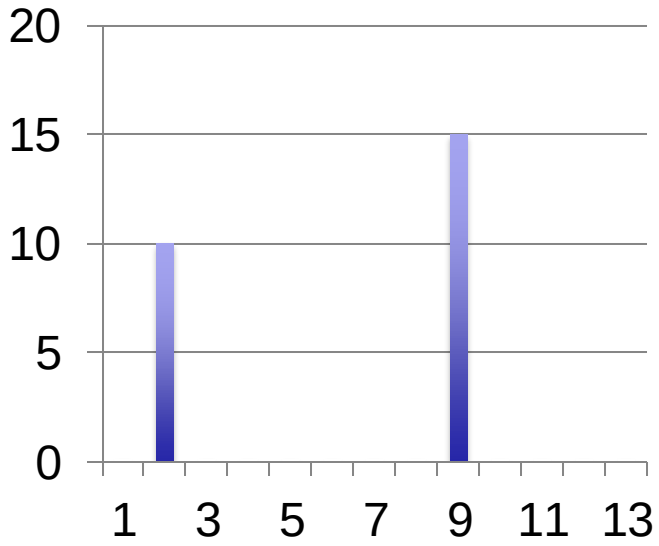
Sparse Approximations

- Goal: approximate a high-dimensional vector x by x' that is sparse, i.e., has few non-zero elements
- Formally: for an n-dimensional vector x, find x' such that
  - x' is k-sparse, i.e., $||x'||_0 \leq k$
  - $||x-x'||_p \leq C\ Err_p^k(x)$

    where

    $Err_p^k(x) = \min_{x''\ k\text{-sparse}} ||x-x''||_p$

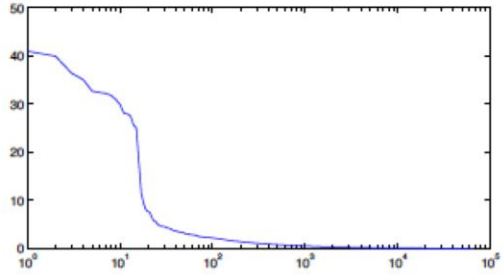




Applications of sparse approximations

- Compression (image, audio, video,….)

Original



Magnitudes of wavelet coefficients

top 10% coefficients

top 3% coefficients

top 1% coefficients

- Heavy hitters, trends, etc
- …

Compressed Sensing

[Candes-Tao; Candes-Romberg-Tao; Donoho]

- A.k.a. compressive sensing or compressive sampling
- Signal acquisition/processing framework:
  - Want to acquire a signal $x=[x_1 \ldots x_n]$
  - Acquisition proceeds by computing Ax+noise of dimension $m<<n$
    - We ignore measurement noise in this talk. But note that x does not have to be k-sparse
  - From Ax we want to recover an approximation x^* of x
  - Method: solve the following program:

$$\text{minimize } ||x^*||_1$$
$$\text{subject to } Ax^*=Ax$$

(other methods available, more later)

  - Guarantee: for some $C>1$

$$\text{(L1/L1)} \quad ||x-x^*||_1 \leq C \min_{k\text{-sparse } x''} ||x-x''||_1$$
$$\text{(L2/L1)} \quad ||x-x^*||_2 \leq C \min_{k\text{-sparse } x''} ||x-x''||_1 / k^{1/2}$$

using $m=O(k \log(n/k))$

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

Application: Single pixel camera

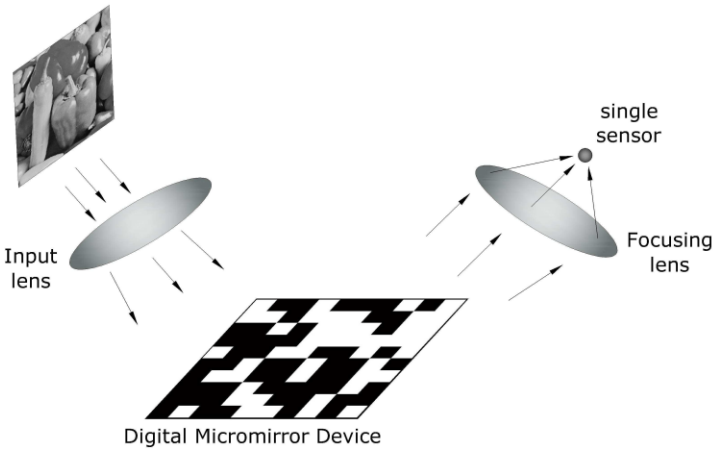
[Wakin, Laska, Duarte, Baron, Sarvotham, Takhar, Kelly, Baraniuk'06,…]

- Measurement:
  - Image x reflected by a mirror a (pixels randomly off and on)
  - The reflected rays are aggregated using lens
  - The sensor receives ax
- Measurement process repeated k times → sensor receives Ax+noise
- Now we want to recover the image from the measurements

Many other architectures, e.g.,

[Fergus-Torralba-Freeman],

[Uttam-Goodman-Neifeld-Kim-John-Kim-Brady], etc



Connections

- Approximation theory
- Screening experiments/group testing
- Statistical model selection (L1 min)
- Estimating Fourier coefficients
- Linear sketching (Andrew's talk)
- Finite rate of innovation
- …

CS on CS insights

- Computer science on compressive sensing
  - Efficient (sub-linear) algorithms:
    - Sparse matrices/hashing
    - Fourier sampling
  - Explicit constructions of matrices
- Compressive sensing on computer science
  - Fast Johnson-Lindenstrauss Transform via RIP property [Ailon-Liberty'11, Krahmer-Ward'11]
  - Other: breaking privacy [Dwork,McSherry,Talwar'07]

Compressive sensing techniques (in one slide)

- A matrix A satisfies RIP of order k with constant δ if for any k-sparse vector Δ we have

$$(1-\delta) \| \Delta \|_2 \leq \|A \Delta \|_2 \leq (1+\delta) \| \Delta \|_2$$

- Theorem: (O(k), δ)-RIP implies that if x* is a solution to the L1 minimization problem, then

$$\text{L2/L1:} \quad \|x-x^*\|_2 \leq C \min_{\text{k-sparse } x''} \|x-x''\|_1 / k^{1/2}$$

- Theorem: If each entry of A is i.i.d. as N(0,1), and $m=\Theta(k \log(n/k))$, then A satisfies (k, δ)-RIP for some δ w.h.p.
  - Proof: via Johnson-Lindenstrauss theorem

- Theorem: If each row of A is chosen i.i.d. from the rows of the Fourier (or Hadamard) matrix, and $m=\Theta(k \log^c n)$ then A satisfies (k, δ)-RIP w.h.p.

Performance

Paper	R/ D	Sketch length	Encode time	Column sparsity	Recovery time	Approx
[CRT'04]	D	k log(n/k)	nk log(n/k)	k log(n/k)	n^c	l2 / l1
[RV'05]	D	k $\log^c$ n	n log n	k $\log^c$ n	n^c	l2 / l1

"Deterministic" guarantee (one matrix A works for all x)

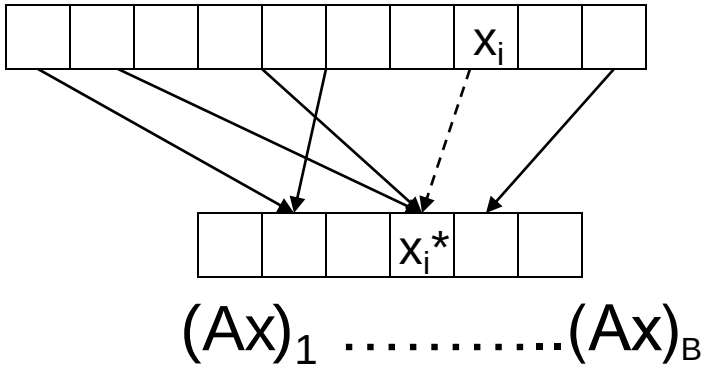
Scale:

Excellent	Very Good	Good	Fair

Linear sketching via sparse binary matrices

- Matrix view:
  - A Bxn matrix A, with one 1 (or +/-1) per column
  - The i-th column has 1 at position h(i), where h(i) be chosen uniformly at random from {1…B}
- Hashing view:
  - h hashes coordinates into "buckets" $(Ax)_1 \ldots (Ax)_B$
  - Ax adds up the coordinates in each bucket
- Large coefficients do not collide with constant probability if B=O(k)
- Repeat log n times

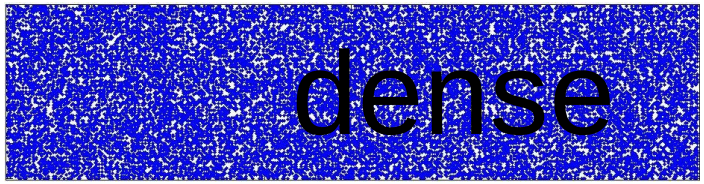
0 0 1 0 0 1 0
1 0 0 0 1 0 0
0 1 0 0 0 0 1
0 0 0 1 0 0 0



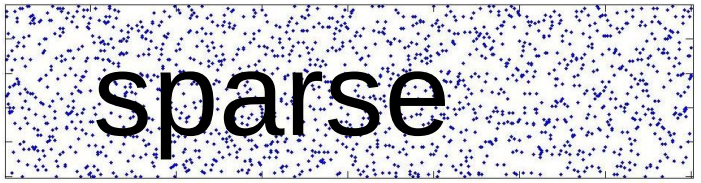
Charikar-Chen-FarachColton'02, Estan-Varghese'03, Cormode-Muthukrishnan'04, Gilbert-Strauss-Vershynin-Tropp'06, Berinde-Gilbert-Indyk-Karloff-Strauss'08, Sarvotham-Baron-Baraniuk'06,'08, Lu-Montanari- Prabhakar'08, Akcakaya-Tarokh'11….

Performance

Paper	R/ D	Sketch length	Encode time	Column sparsity	Recovery time	Approx
[CCF'02], [CM'06]	R	k log n	n log n	log n	n log n	l2 / l2
	R	k $\log^c$ n	n $\log^c$ n	$\log^c$ n	k $\log^c$ n	l2 / l2
[CM'04]	R	k log n	n log n	log n	n log n	l1 / l1
	R	k $\log^c$ n	n $\log^c$ n	$\log^c$ n	k $\log^c$ n	l1 / l1
[CRT'04] [RV'05]	D	k log(n/k)	nk log(n/k)	k log(n/k)	n^c	l2 / l1
	D	k $\log^c$ n	n log n	k $\log^c$ n	n^c	l2 / l1
[GSTV'06] [GSTV'07]	D	k $\log^c$ n	n $\log^c$ n	$\log^c$ n	k $\log^c$ n	l1 / l1
	D	k $\log^c$ n	n $\log^c$ n	k $\log^c$ n	k^2 $\log^c$ n	l2 / l1
[BGIKS'08]	D	k log(n/k)	n log(n/k)	log(n/k)	n^c	l1 / l1
[GLR'08]	D	k $\log n^{\log\log\log n}$	kn^{1-a}	n^{1-a}	n^c	l2 / l1
[NV'07], [DM'08], [NT'08], [BD'08], [GK'09], …	D	k log(n/k)	nk log(n/k)	k log(n/k)	nk log(n/k) * log	l2 / l1
	D	k $\log^c$ n	n log n	k $\log^c$ n	n log n * log	l2 / l1
[IR'08], [BIR'08],[BI'09]	D	k log(n/k)	n log(n/k)	log(n/k)	n log(n/k)* log	l1 / l1
[GLPS'10]	R	k log(n/k)	n $\log^c$ n	$\log^c$ n	k $\log^c$ n	l2 / l2
Open problem 1:	D	k logn(n/k)			n polylog n	l2/l1



vs.



- Restricted Isometry Property (RIP)

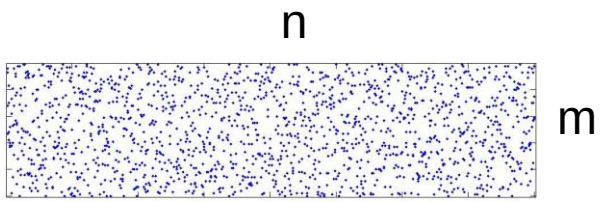
$$\Delta \text{ is } k\text{-sparse } \Rightarrow ||\Delta||_2 \leq ||A\Delta||_2 \leq C\,||\Delta||_2$$

- Consider $m \times n$ 0-1 matrices with d ones per column
- Do they satisfy RIP ?
  - No, unless $m=\Omega(k^2)$ [Chandar'07]
- However, they can satisfy the following RIP-1 property [Berinde-Gilbert-Indyk-Karloff-Strauss'08]:

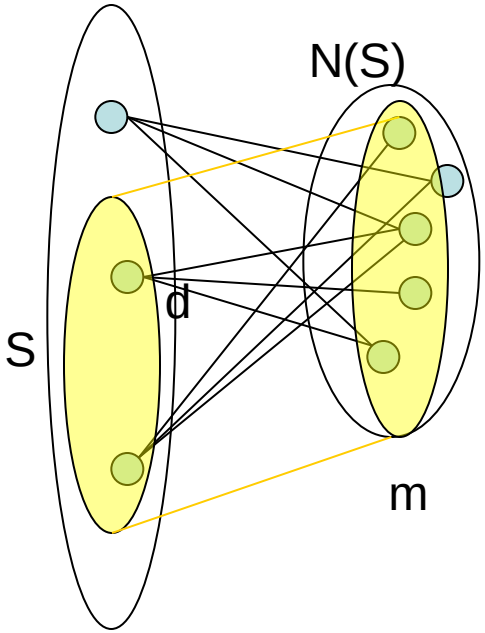
$$\Delta \text{ is } k\text{-sparse } \Rightarrow d\,(1-\varepsilon)\,||\Delta||_1 \leq ||A\Delta||_1 \leq d||\Delta||_1$$

  - Sufficient (and necessary) condition: the underlying graph is a $(k, d(1-\varepsilon/2))$-expander
- Implies L1/L1 guarantee for L1 minimization (and other algorithms)

Expanders



- A bipartite graph is a $(k,d(1-\varepsilon))$-expander if for any left set S, $|S| \leq k$, we have $|N(S)| \geq (1-\varepsilon)d|S|$
- Objects well-studied in theoretical computer science and coding theory
- Probabilistic construction yields
  $m=O(k \log (n/k))$
- RIP-1 yields:



[BGIKS'08]	D	k log(n/k)	n log(n/k)	log(n/k)	n^c	l1 / l1

- Coding theory (LDPC) handles exactly k-sparse case [Xu-Hassibi'07, Indyk'08, Jafarpour-Xu-Hassibi-Calderbank'08]

Explicit constructions

- Explicit constructions of expanders: m=k quasipolylog n
  [….,Guruswami-Umans-Vadhan'07]
- Yields matrix with RIP-1 property and k quasipolylog n rows
- Alternate approach: Explicit Lp sections of L1 [Guruswami-Lee-Razborov'08; Karnin'11]
  - Stronger guarantee (Lp/L1)
  - Higher number of measurements
- Recent: explicit construction of RIP-2 matrix with $m=k^{2-\varepsilon}$ [Bourgain-Dilworth-Ford-Konyagin-Kutzarova'11]
- Open problem 2: better bounds ?

CS on CS insights

- Efficient (sub-linear) algorithms:
  - Sparse matrices/hashing ✓
  - Fourier sampling
- Explicit constructions of matrices ✓

- Fast Johnson-Lindenstrauss Transform via RIP property [Ailon-Liberty'11, Krahmer-Ward'11]

Fourier sampling

R/D	Sketch length	Encode time	Column sparsity	Recovery time	Approx
D	$k \log^c n$	$n \log n$	$k \log^c n$	n^c	l2 / l1

- $\Theta(k \log^c n)$ Fourier measurements of x suffice to recover a k-sparse approximation to x
- Equivalently, can recover a k-sparse approximation in the Fourier basis to x from $\Theta(k \log^c n)$ samples of x
- Fourier sampling:
  - Boolean cube (Hadamard Transform): [KM] (cf. [GL])
  - Complex FT: [Mansour'92, GGIMS'02, AGS'03, GMS'05, Iwen'10, Akavia'10,HIKP'12]
- Best time/sample bound [HIKP'12]: $O(k \log(n) \log(n/k))$
- Open problem 3: $O(k \log(n))$ time and/or sample bound

CS on CS insights

- Efficient (sub-linear) algorithms:
  - Sparse matrices/hashing ✓
  - Fourier sampling ✓
- Explicit constructions of matrices ✓

- Fast Johnson-Lindenstrauss Transform via RIP property [Ailon-Liberty'11, Krahmer-Ward'11]

JL lemma

- Example statement:
  - Let A be a $k \times n$ matrix, with i.i.d. entries chosen uniformly at random from $\{-1,1\}$
  - Then, for any $x \in R^n$, we have

$$\Pr[\, | \, ||Ax||^2 - k||x||^2 \, | > \varepsilon k||x||^2] \leq \exp(-C \, \varepsilon^2 \, k)$$

- Powerful hammer
- The time to compute Ax: $O(nk)$

Fast Johnson-Lindenstrauss Transform

[Ailon-Chazelle'06]

- A structured matrix A, with guarantees as before
- The time to compute Ax:

$$O(n \log n) + k^c$$

  - [AC] showed c=3
  - Several improvements
- Idea: compute

$$Ax = P\ H\ D\ x$$

where:

  - D: diagonal matrix, with $D_{ii} \in \{-1,1\}$ i.i.d.
  - H: normalized Hadamard matrix
    - Entries in $\{-1/n^{1/2}, 1/n^{1/2}\}$
    - Columns orthogonal
  - P: a sparse projection matrix (several options possible)

RIP implies JL

[Ailon-Liberty'11, Krahmer-Ward'11]:

- Theorem: Consider $A = B D$ where B has RIP property of order k (e.g., can choose B = random rows from H).

  Then A satisfies JL lemma with error probability at most $\exp(-k)$
  - Running time: $O(n \log n)$
  - But dimension $O(k * \log^{O(1)} n)$ as opposed to $O(k)$
- Open problem 4: JL transform with $O(n * \log^{O(1)} n)$ time and dimension $O(k)$

  (this would actually solve problem 1 as well !)

Conclusions

- Many insights in both directions
- Many interesting open problems
- Some resources:
  - Compressive sensing:

    J. A. Tropp, S. J. Wright, “Computational Methods for Sparse Solution of Linear Inverse Problems”, Proceedings of IEEE, June 2010.
  - Sparse (+explicit) matrices:

    A. Gilbert, P. Indyk, “Sparse recovery using sparse matrices”, Proceedings of IEEE, June 2010.
  - Fast JL transform

    N. Ailon, B. Chazelle, “Faster Dimension Reduction”, CACM, 2010.