One-bit Compressed Sensing: Provable Support and Vector Recovery

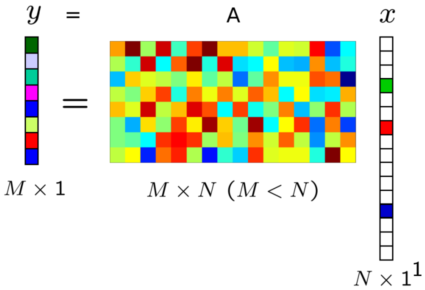
Praneeth Netrapalli

The University of Texas at Austin

Joint work with **Sivakant Gopi**, **Prateek Jain** and **Aditya Nori**

Jun 17, 2013

Compressed Sensing



- Goal: Reconstruct a sparse signal using very few linear measurements
- Tremendous amount of work in the last decade
- $O(k \log n)$ measurements to reconstruct k-sparse signals in $\mathbb{R}^n$

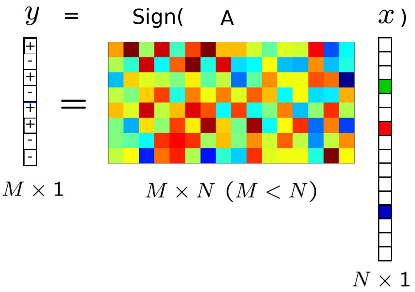
[1] http://lions.epfl.ch/research

Quantization

- Measurements up to infinite precision - not practical
  - $y_j = -1.010001011110$
- Arbitrary quantization does not work well [BB08]
  - $y_j = -1.01$
- Extreme quantization - single bit measurements
  - $y_j = -1$

One-bit Compressed Sensing [BB08]

Goal: Reconstruct a sparse signal using *signs* of very few linear measurements



- Formally, given y and A, recover x
- Motivation:
  - Captures extreme quantization
  - Easy to implement
  - Robust to noise

Frameworks

Exact recovery not possible:

For example, $y = \mathrm{Sign}\left(Ax\right) = \mathrm{Sign}\left(A\left(cx\right)\right),\ \forall c > 0$

Frameworks

- Support recovery: recover $\widehat{x}$ such that

$$\mathrm{Supp}\left(x\right) = \mathrm{Supp}\left(\widehat{x}\right)$$

- ϵ-approximate recovery[a]: recover k-sparse $\widehat{x}$ such that

$$\left\| \widehat{x} - \frac{x}{\|x\|} \right\| < \epsilon$$

[a] $\|\cdot\|$ refers to two norm

Metrics

- Measurement (or sample) complexity: dimension of y
- Computational complexity: time taken by the recovery algorithm
- Universality: use the same measurement matrix A for all sparse vectors x
  - critical for many applications e.g., single pixel camera

Our Results – Support Recovery

Algorithm	[HB11]	UFF	Expanders

Our Results – Support Recovery

Algorithm	[HB11]	UFF	Expanders
Universal	No	Yes	Yes

Our Results – Support Recovery

Algorithm	[HB11]	UFF	Expanders
Universal	No	Yes	Yes
-ve entries allowed?	Yes	No	Yes

Our Results – Support Recovery

Algorithm	[HB11]	UFF	Expanders
Universal	No	Yes	Yes
-ve entries allowed?	Yes	No	Yes
# measurements	$O(k \log n)$	$O\left(k^2 \log n\right)$	$O\left(k^3 \log n\right)$

Our Results – Support Recovery

Algorithm	[HB11]	UFF	Expanders
Universal	No	Yes	Yes
-ve entries allowed?	Yes	No	Yes
# measurements	$O(k \log n)$	$O(k^2 \log n)$	$O(k^3 \log n)$
Running Time	$O(n \log n)$	$O(nk \log n)$	$O(nk \log n)$

- *First universal* measurement schemes
- Open problem: $O(k \log n)$ universal measurement scheme not known

Union Free Family

Definition (k-Union Free Family)

Sets $\mathcal{F} := \{B_1, \cdots, B_n\}$*:*

Union Free Family

Definition (k-Union Free Family)

Sets $\mathcal{F} := \{B_1, \cdots, B_n\}$*:*

$$B_{i_0} \not\subseteq B_{i_1} \cup \cdots \cup B_{i_k},$$

for all distinct $B_{i_0}, B_{i_1}, \cdots, B_{i_k} \in \mathcal{F}$*.*

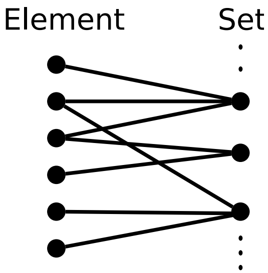
Union Free Family

Definition (k-Union Free Family)

Sets $\mathcal{F} := \{B_1, \cdots, B_n\}$*:*

$$B_{i_0} \not\subseteq B_{i_1} \cup \cdots \cup B_{i_k},$$

for all distinct $B_{i_0}, B_{i_1}, \cdots, B_{i_k} \in \mathcal{F}$.



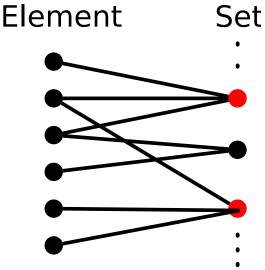
Union Free Family

Definition (k-Union Free Family)

Sets $\mathcal{F} := \{B_1, \cdots, B_n\}$*:*

$$B_{i_0} \not\subseteq B_{i_1} \cup \cdots \cup B_{i_k},$$

for all distinct $B_{i_0}, B_{i_1}, \cdots, B_{i_k} \in \mathcal{F}$*.*



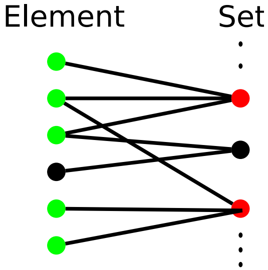
Union Free Family

Definition (k-Union Free Family)

Sets $\mathcal{F} := \{B_1, \cdots, B_n\}$*:*

$$B_{i_0} \not\subseteq B_{i_1} \cup \cdots \cup B_{i_k},$$

for all distinct $B_{i_0}, B_{i_1}, \cdots, B_{i_k} \in \mathcal{F}$*.*



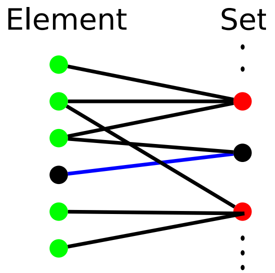
Union Free Family

Definition (k-Union Free Family)

Sets $\mathcal{F} := \{B_1, \cdots, B_n\}$*:*

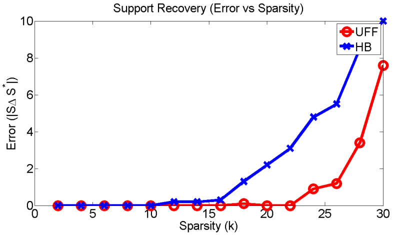
$$B_{i_0} \not\subseteq B_{i_1} \cup \cdots \cup B_{i_k},$$

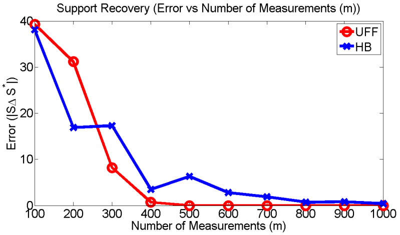
for all distinct $B_{i_0}, B_{i_1}, \cdots, B_{i_k} \in \mathcal{F}$*.*



- Well studied combinatorial objects
- Use their structure in the measurement scheme *and* recovery algorithm
- Works for $x \geq 0$
  - Another algorithm based on expanders, which works for any x

Experiments





ϵ-approximate recovery

- Recall: Need to recover $\widehat{x}$ such that $\left\| \widehat{x} - \frac{x}{\|x\|} \right\| < \epsilon$
- Plan and Vershynin [PV11, PV12]: $O\left(\epsilon^{-5}\right)$ measurements
  - hard thresholding and soft thresholding based approaches
- Our result: $\widetilde{O}\left(\epsilon^{-1}\right)$ measurements
- Key idea: Combine optimal results from compressed sensing and learning halfspaces

Our Results – Approximate Recovery

Algorithm	[PV11, PV12]	Two-stage	S-Approx

Our Results – Approximate Recovery

Algorithm	[PV11, PV12]	Two-stage	S-Approx
Universal	Yes	Yes	Yes

Our Results – Approximate Recovery

Algorithm	[PV11, PV12]	Two-stage	S-Approx
Universal	Yes	Yes	Yes
# measurements (m)	$\frac{k}{\epsilon^5} \log^2 \frac{n}{k}$	$\frac{k}{\epsilon} \log \frac{n}{k}$	$k^3 \log \frac{n}{k} + \frac{k}{\epsilon}$

Our Results – Approximate Recovery

Algorithm	[PV11, PV12]	Two-stage	S-Approx
Universal	Yes	Yes	Yes
# measurements (m)	$\frac{k}{\epsilon^5} \log^2 \frac{n}{k}$	$\frac{k}{\epsilon} \log \frac{n}{k}$	$k^3 \log \frac{n}{k} + \frac{k}{\epsilon}$
Running Time	$\frac{kn}{\epsilon^6} \log \frac{n}{k}$	$\frac{kn}{\epsilon^5} \log \frac{n}{k}$	$\frac{k^5 n}{\epsilon^5} \log \frac{n}{k}$

- Two-stage: *Near optimal* measurement complexity

Measurement Scheme and Algorithm

$$y = Sign(\overbrace{A_2 A_1}^{A} x)$$

Random Gaussian Matrix (A_2)

Compressed Sensing Matrix (A_1)

Measurement Scheme and Algorithm

$$y = Sign(\overbrace{\underset{\substack{\text{Random}\\ \text{Gaussian}\\ \text{Matrix}}}{A_2}\underset{\substack{\text{Compressed}\\ \text{Sensing}\\ \text{Matrix}}}{A_1}}^{A} x)$$

Algorithm

1. Linear programming: Obtain $\widehat{z}$ such that $y = \mathrm{Sign}\left(A_2\widehat{z}\right)$

Measurement Scheme and Algorithm

$$y = Sign(\overbrace{\underbrace{A_2}_{\text{Random Gaussian Matrix}}\underbrace{A_1}_{\text{Compressed Sensing Matrix}}}^{A} x)$$

Algorithm

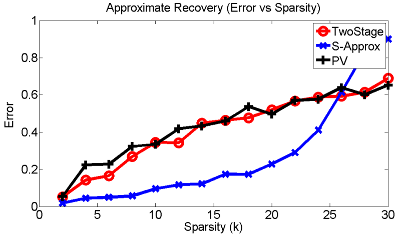
1. Linear programming: Obtain $\hat{z}$ such that $y = \mathrm{Sign}\left(A_2 \hat{z}\right)$
2. Solve CS problem: $\hat{z} = A_1 x + e$ (GraDeS [GK09])

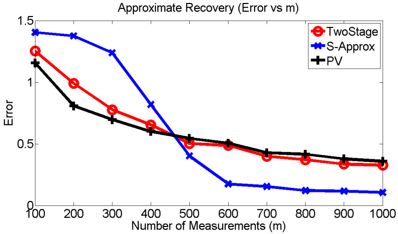
Proof Outline

$$y = Sign(A_2 z), \qquad A_2 z = A_1 x$$

1. Robust CS: sufficient to obtain $\widehat{z}$ such that $\|\widehat{z} - z\| < C\epsilon$.
   - requires $O(k \log n)$ measurements
2. Obtain $\widehat{z}$: requires $O\left(\frac{k \log n}{\epsilon} \log\left(\frac{k \log n}{\epsilon}\right)\right)$ random Gaussian measurements.

Experiments





Summary

- One-bit Compressed Sensing
  - captures extreme quantization
  - several other motivations
- Support recovery
  - *First universal* measurement schemes using UFF and expanders
  - UFF: $O\left(k^2 \log n\right)$ measurements but for $x \geq 0$
    - Can be used in other settings?
  - Expanders: $O\left(k^3 \log n\right)$ measurements for any x
  - Open question: achieving $O\left(k \log n\right)$ sample complexity
- ϵ-approximate recovery
  - *Near optimal* sample complexity with ϵ
  - Support recovery based algorithm - works well empirically

References

Petros Boufounos and Richard G. Baraniuk.
1-bit compressive sensing.
In *CISS*, pages 16–21, 2008.

Rahul Garg and Rohit Khandekar.
Gradient descent with sparsification: an iterative algorithm for sparse recovery with restricted isometry property.
In *ICML*, 2009.

Jarvis Haupt and Richard G. Baraniuk.
Robust support recovery using sparse compressive sensing matrices.
In *CISS*, pages 1–6, 2011.

Y. Plan and R. Vershynin.
One-bit compressed sensing by linear programming.
arXiv preprint arXiv:1109.4299, 2011.

Yaniv Plan and Roman Vershynin.
Robust 1-bit compressed sensing and sparse logistic regression: A convex programming approach.
CoRR, abs/1202.1212, 2012.