An Algorithm for Exact Super-resolution and Phase Retrieval

Yuxin Chen, Yonina Eldar and Andrea Goldsmith

Stanford University, Technion

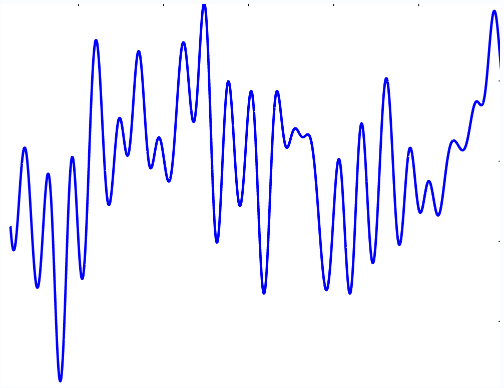
Sparse Representation

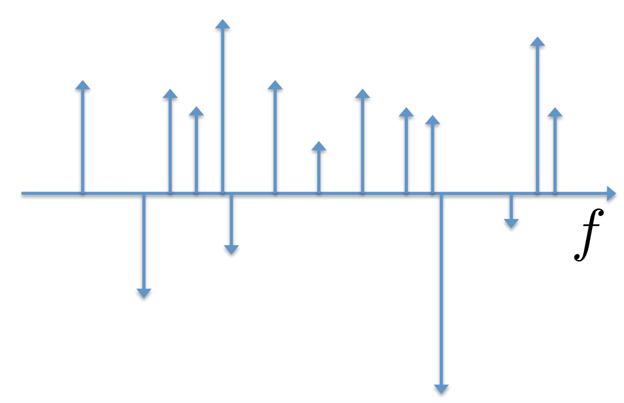
- Fourier representation

$$x(\boldsymbol{t}) = \sum_{i=1}^{r} d_i e^{j2\pi\langle \boldsymbol{t}, \boldsymbol{f}_i \rangle}$$

($\boldsymbol{f}_i$: frequencies, d_i : amplitudes, r: model order)

- **Sparsity**: nature is sparse
- **Goal**: identify the spikes from frequency samples

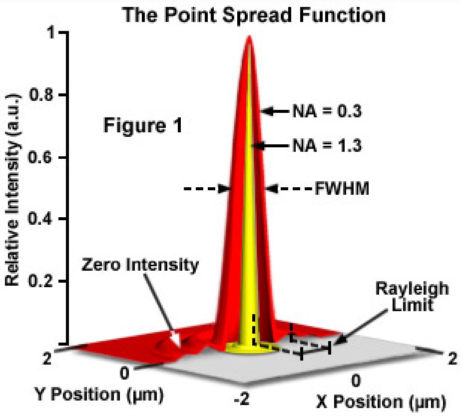




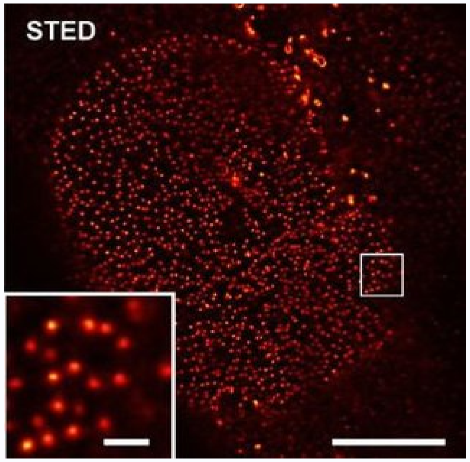
Application in Imaging

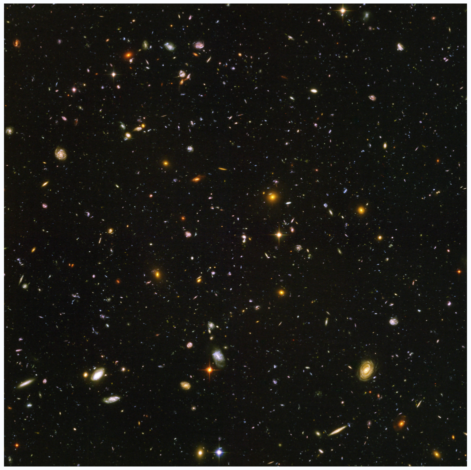
- Time-sparse signals

$$z\left(\boldsymbol{t}\right)=\sum_{i=1}^{r}d_i\delta(t-\boldsymbol{t}_i)$$



- Imaging Resolution:
  - limited by the point spread function of the imaging system

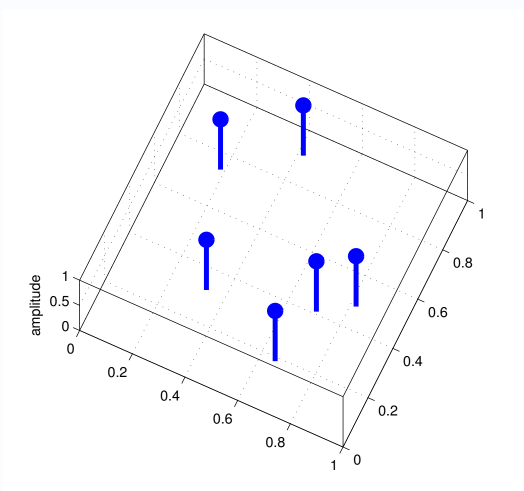




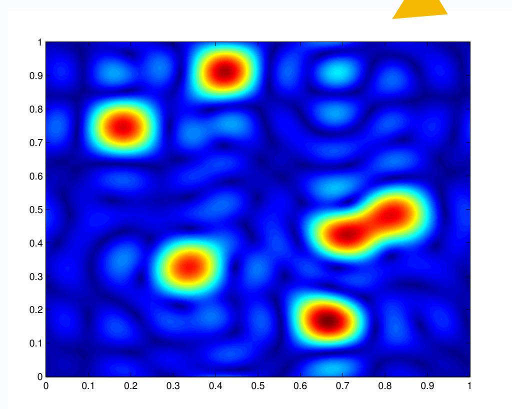
Super-Resolution

- obtain low-pass signals → extrapolate to high frequencies

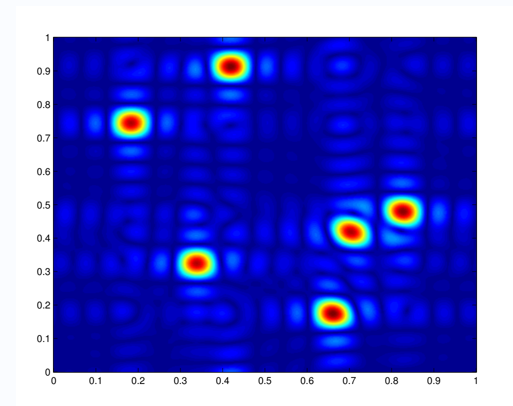
sparse signal → Low-pass filter → **low-end spectrum**



(a) Ground Truth



(b) Low Resolution Image

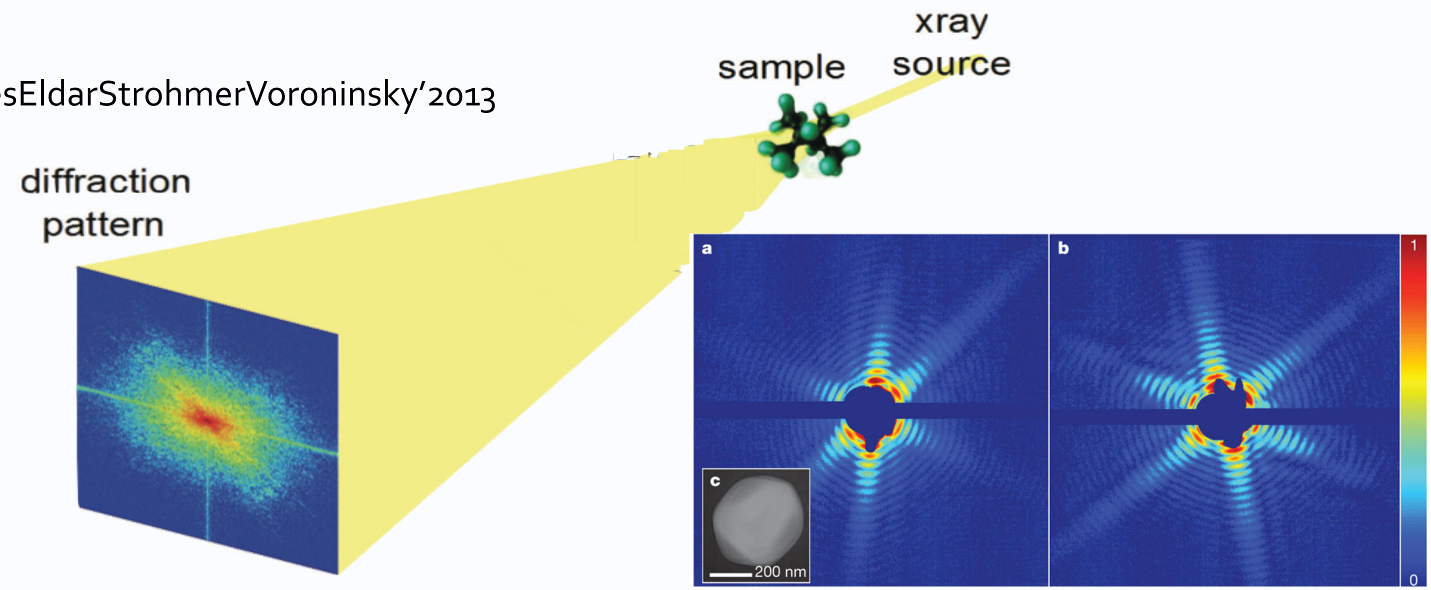


(c) Super-Resolution

(Figure: ChenChi'2013)

Phase Retrieval

Figure: CandesEldarStrohmerVoroninsky'2013

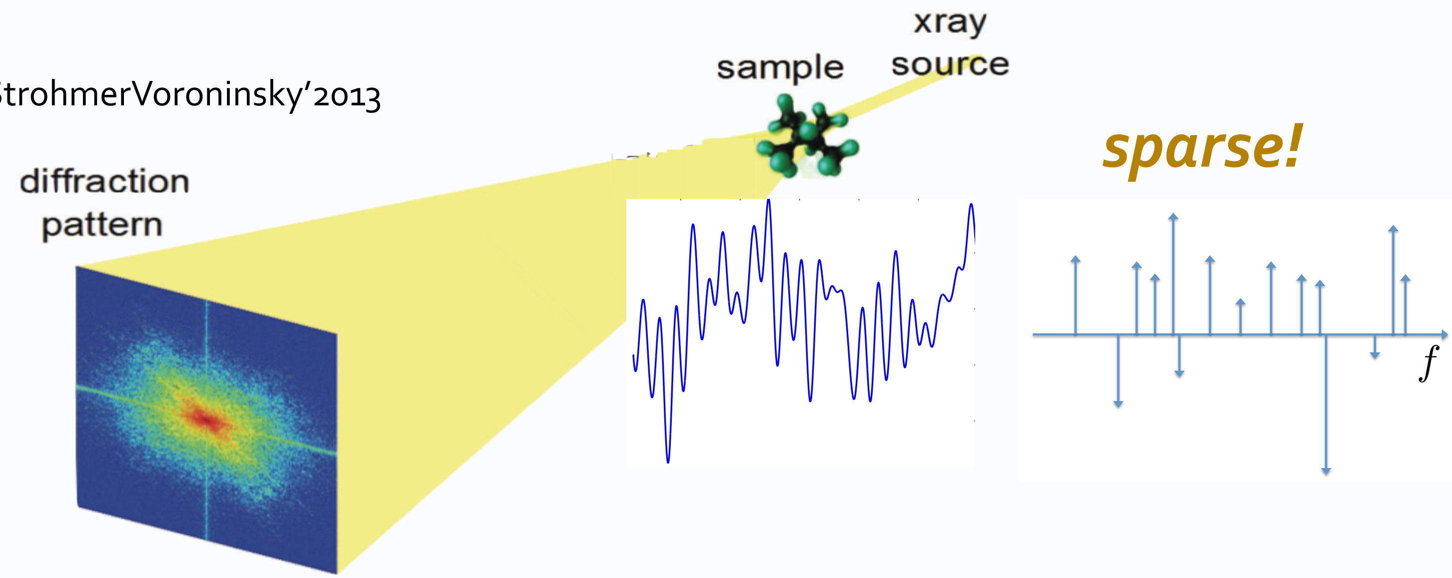


signal → Fourier transform → amplitude → Fourier intensity

Recover?

Phase Retrieval from Low-End Spectrum

CandesEldarStrohmerVoroninsky'2013

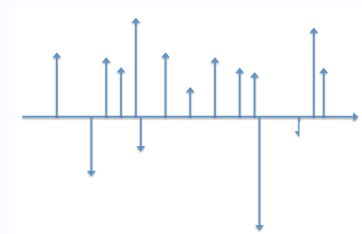


sparse signal → Fourier transform → **low-pass filter** → amplitude → **low-end** Fourier intensity

Recover?

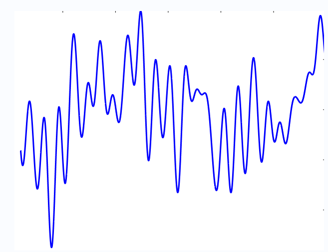
Problem Formulation

true signal: $$x(t) = \sum_{l=1}^{r} a_l \delta(t - t_l),$$



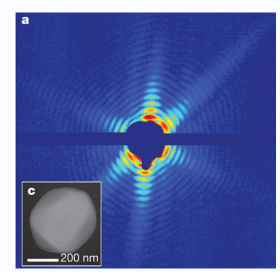
Fourier transform

Fourier response: $$\hat{x}[k] = \sum_{l=1}^{r} a_l e^{-j2\pi k t_l}$$



low-pass + amplitude^2

low-end Fourier intensity: $$y[k] := |\hat{x}[k]|^2, \qquad \underbrace{-m_c \le k < m_c}_{\text{low-pass}}$$



Goal: recover $x(t)$ from $y[k]$ $(-m_c \le k \le m_c)$

A *Small* Sample of Prior Work

Problem: recover $x(t)$ from $y[k]$ $(-m_c \leq k \leq m_c)$

$$\boxed{y[k] := |\hat{x}[k]|^2}$$

- Non-convex / Non-linear

- Prior algorithm:
  - **SDP** (Candes et al): works under *random masks*
  - **GESPAR** (Shechtman et al): empirically fast but no provable guarantees
  - **Combinatorial** (Jaganathan et al): need *non-uniform* support
  - **Uniqueness**: Ohlsson et al, Ranieri et al, …

Our Goal

Problem: recover $\boldsymbol{x}(\boldsymbol{t})$ from $\boldsymbol{y}[\boldsymbol{k}]$ $(-\boldsymbol{m}_{\mathrm{c}} \leq \boldsymbol{k} \leq \boldsymbol{m}_{\mathrm{c}})$

$$\boxed{y[k] := |\hat{x}[k]|^2}$$

- Develop a ***tractable*** algorithm that
  - **Exact recovery (*<u>under minimal assumptions!</u>*)**
  - **Deterministic guarantees**
    - Prior work mostly on *with-high-prob* guarantees
  - **Works for both continuous and discrete domains**
    - Basis mismatch issue: Chi et al 2011

Spectral Sparsity of Amplitude Samples

$$\boxed{y[k] := |\hat{x}[k]|^2}$$

***r*-sparse signal:** $\hat{x}[k] = \sum_{l=1}^{r} a_l e^{-j2\pi k t_l}$

expand

Amplitude²: $y[k] := |\hat{x}(k)|^2 = \underbrace{\sum_{i=1}^{r}\sum_{l=1}^{r} \underbrace{a_i a_l^*}_{a_{i,l}} \exp\left(-j2\pi \underbrace{(t_i - t_l)}_{t_{i,l}} k\right)}_{(r^2-r+1)\text{-sparse}}$

***(r²-r+1)*-sparse**

Observation: amplitude² is spectrally sparse

Matrix Pencil Approach

Observation: amplitude^2 $y[k]$ is spectrally sparse

- Conventional Harmonic Retrieval:
  - Can recover $y[k]$ from $\mathbf{2(r^2\text{-}r\text{+}1)}$ samples!

- Example: Matrix Pencil Approach

$$\boldsymbol{Y} = \begin{bmatrix} Y[0] & Y[1] & \cdots & Y[m_c] \\ Y[1] & Y[2] & \cdots & Y[m_c+1] \\ \vdots & \vdots & \cdots & \vdots \\ Y[m_c-1] & Y[m_c] & \cdots & Y[2m_c-1] \\ Y[m_c] & Y[m_c+1] & \cdots & Y[2m_c] \end{bmatrix}$$

$\boldsymbol{Y_1}$

$\boldsymbol{Y_2}$

Examine the spectrum of $\boldsymbol{Y_1^{-1}Y_2}$

Hankel structure

Does Matrix Pencil suffice?

Observation: amplitude^2 $y[k]$ is spectrally sparse

- Matrix Pencil Approach
  - retrieve all spectral spikes of $y[k]$?

$$y[k] := |\hat{x}(k)|^2 = \sum_{i=1}^{r} \sum_{l=1}^{r} \underbrace{a_i a_l^*}_{a_{i,l}} \exp\Big(-j2\pi \underbrace{(t_i - t_l)}_{t_{i,l}} k \Big)$$

- What have we recovered so far?

$\{a_i a_l^* : i, l\}$ and $\cancel{\{t_i - t_l : i, l\}}$ $\{|t_i - t_j| : i \neq j\}$

Unlabeled Sets!!

Recover amplitudes a_i via Sorting

Given: *unlabeled set* $\{a_i a_l^* : i, l\}$

Want: $\{a_i : i\}$

$$\boxed{\text{suppose} \quad |a_1| > |a_2| > \cdots > |a_r|}$$

Question: if we know $|a_1|$, how to find $|a_2|$?

Answer : the largest in $\{|a_i a_l| : i \neq l\}$ is $\boxed{|a_1 a_2|}$

$$|a_1| + |a_1 a_2| \longrightarrow |a_2|$$

Recover amplitudes a_i via Sorting

Given: *unlabeled set* $\{a_i a_l^* : i, l\}$

Want: $\{a_i : i\}$

$\boxed{\textbf{suppose} \quad |a_1| > |a_2| > \cdots > |a_r|}$

Question: if we know $|a_1|$ and $|a_2|$, how to find $|a_3|$?

Answer : the largest in $\{|a_i a_l| : i \neq l\} \setminus \{|a_1 a_2|\}$ is $\boxed{|a_1 a_3|}$

$$|a_1| + |a_1 a_3| \longrightarrow |a_3|$$

Recover amplitudes a_i via Sorting

Given: *unlabeled set* $\{a_i a_l^* : i, l\}$

Want: $\{a_i : i\}$

$\boxed{\textbf{suppose} \quad |a_1| > |a_2| > \cdots > |a_r|}$

Continue…

Conclusion : *when* $|a_1|$ *is known,* then we can easily get all $|a_i|$

Question: how to retrieve $|a_1|$?

Retrieve $|a_1|$

suppose $|a_1| > |a_2| > \cdots > |a_r|$

Intuition: $|a_1 a_2| + |a_1 a_3| + |a_2 a_3| \Longleftrightarrow |a_1| + |a_2| + |a_3|$

What is the largest in $\{|a_i a_l| : i \neq l\}$? **Answer:** $|a_1 a_2|$

2nd largest? **Answer:** $|a_1 a_3|$

3nd largest? **Answer:** *not necessarily* $|a_2 a_3|$

But $|a_2 a_3|$ must be among the *r*+1 largest

$\Rightarrow$ *search over all possibilities (still tractable)!*

Retrieve t_i (similarly arg(a_i))

Given: $|a_1|, |a_2|, \cdots, |a_r|$ and $\{|t_i - t_j| : i \neq j\}$

Want: $t_1, t_2, \cdots, t_r$

Step 1. $|a_1|, |a_2|, \cdots, |a_r|$ + $\{|a_i a_l| : i \neq l\}$ ⇒ **labels**

Step 2. $\{|t_i - t_j| : i \neq j\}$ + **labels** ⇒ $|t_i - t_j| (i \neq j)$

Step 3. $|t_i - t_j| (i \neq j)$ ⇒ $t_1, t_2, \cdots, t_r$

Use any graph realization algorithm !

Summary: 2-Stage Algorithm

- Stage 1: Matrix Pencil to retrieve $\underbrace{\{a_i a_l^* : i \neq l\} \{|t_i - t_j| : i \neq j\}}_{\textit{unlabelled}}$

- Stage 2: Recover $t_1, t_2, \cdots, t_r$ and $\{a_i : i\}$
  - Use sorting to retrieve $|a_1|, |a_2|, \cdots, |a_r|$
  - This recovers the "labels"
  - Use graph realization to retrieve $t_1, t_2, \cdots, t_r$

Summary: When does it work?

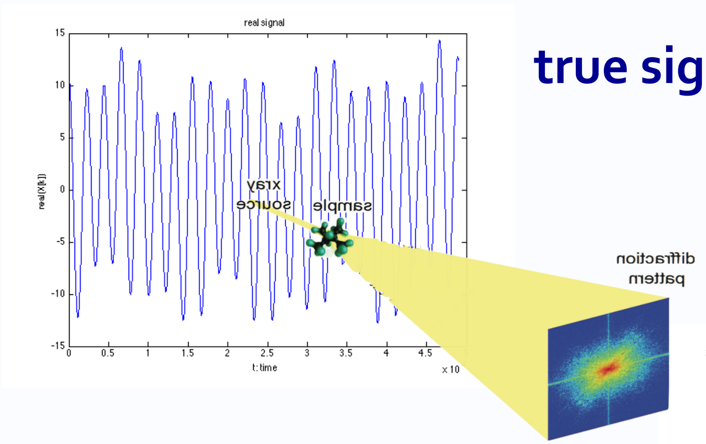
- The proposed algorithm works if

$$|a_1| > |a_2| > \cdots > |a_r| \qquad \textbf{(sorting)}$$

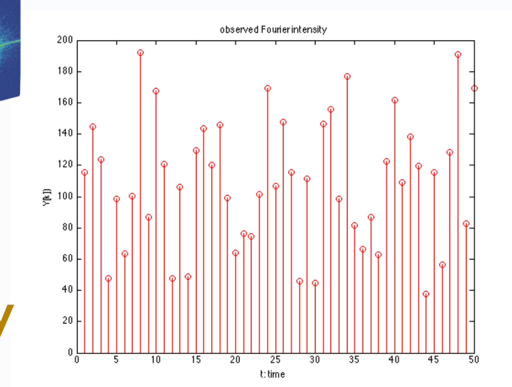
$$\#\ \text{samples}\ \geq 2r^2 - 2r + 3 \qquad \textbf{(matrix pencil)}$$

- Near-Minimal assumptions?

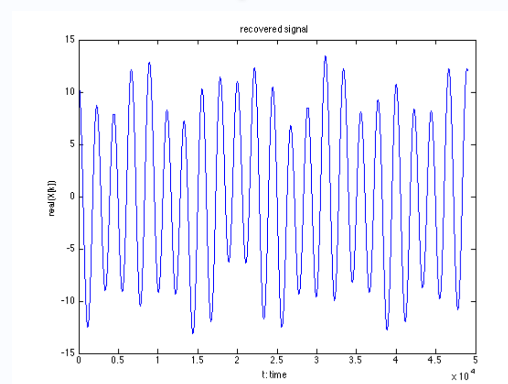
Numerical Example



true signal (5 random spikes)



low-end Fourier intensity



recovered signal

Computational Gap?

- Is it possible to recover from $o(r^2)$ samples (with no further assumptions)?

- Answer: *I don't know…*
- **My conjecture:**
  - *Probably not!*
  - Deep connection to *sparse PCA, planted clique, …*
  - *For these problems, computational gap exists!*

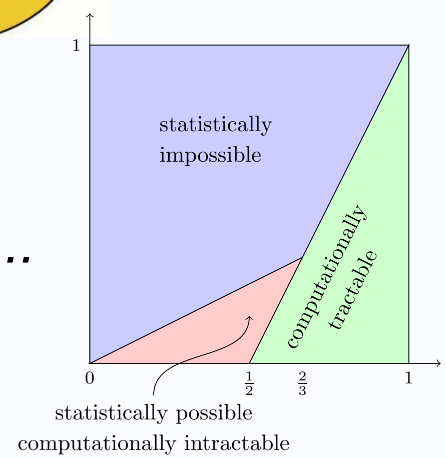


figure:MaWu'2013

Conclusion

- Propose a simple algorithm for PR + SuperRes
  - deterministic recovery guarantees
  - accommodate continuous spike locations
- Question:
  - whether there exists a computational barrier?