Adaptive Compressive Imaging
Using Sparse Hierarchical Learned Dictionaries

Jarvis Haupt
University of Minnesota
Department of Electrical and Computer Engineering

Supported by DARPA
— Motivation —

New Agile Sensing Platforms
A Host of New Agile Imaging Sensors

Photodiode, photomultiplier, hyperspectral imager, etc.

Single-pixel Camera (Rice University)

Original           20%                40% samples         samples

CMOS Separable Transform Image Sensor (Georgia Tech)

DCT Basis Set  Noislet Basis Set

No Compression

23% Compression

48% Compression

72% Compression
– Overview of This Talk –

Fusing Adaptive Sensing and Structured Sparsity in theory and in practice…
– Background –

Sparse Inference and Adaptive Sensing
A Model for Sparsity

Objects of interest are vectors \( x \in \mathbb{R}^n \)

Signal Support: \( S \triangleq \{ i : x_i \neq 0 \} \)

Sparse \( \iff |S| = k \ll n \)
A Sparse Inference Task

Noisy Linear Observation Model:

\[
y = \Phi x + w \quad \left\{ \begin{array}{l}
\Phi \in \mathbb{R}^{m \times n} \\
w \sim \mathcal{N}(0, I_{m \times m})
\end{array} \right.
\]

Support Recovery
Goal: Obtain an (accurate) estimate \( \hat{S} = \hat{S}(y, \Phi) \) of true support \( S \)
A Sparse Inference Task

Noisy Linear Observation Model:

\[ y = \Phi x + w \]

\[ \Phi \in \mathbb{R}^{m \times n} \]
\[ w \sim \mathcal{N}(0, I_{m \times m}) \]

Assume

- "Sensing energy" \( \| \Phi \|_F^2 \) fixed: \( \| \Phi \|_F^2 = R \)
- \( |x_i| \geq \mu \) for all \( i \in S \)

What conditions are necessary/sufficient for \textit{exact} support recovery?
(eg., such that \( P(S \neq \hat{S}) \to 0 \) as \( n \to \infty \))
Exact Support Recovery?

“Point sampling” \( y = x + w \)
(Sensing energy \( R = n \))

Necessary & Sufficient for Exact Support Recovery:

\[
\mu \geq \text{const.} \sqrt{\left( \frac{n}{R} \right) \log n}
\]

“Compressed” Sensing (Genovese, Jin, & Wasserman 2009; Aeron, Saligrama & Zhao, 2010)
Conditions for Exact Support Recovery

"Compressed" Sensing (Genovese, Jin, & Wasserman 2009; Aeron, Saligrama & Zhao, 2010)

Question: Can we do better by exploiting structure, or adaptivity, or both?
# Conditions for Exact Support Recovery

<table>
<thead>
<tr>
<th></th>
<th>Non-structured</th>
<th>Structured</th>
<th>Non-adaptive</th>
<th>Adaptive</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-adaptive</td>
<td>( \mu \geq \text{const.} \sqrt{\left( \frac{n}{R} \right) \log n} )</td>
<td>?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Adaptive</td>
<td>( \mu \geq \text{const.} \sqrt{\left( \frac{n}{R} \right) \log k} )</td>
<td>?</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\( \mathbf{y} = \Phi \mathbf{x} + \mathbf{w} \) \quad \left\{ \begin{array}{l}
\Phi \in \mathbb{R}^{m \times n} \\
w \sim \mathcal{N}(0, I_{m \times m}) \\
\|\Phi\|_F^2 = R
\end{array} \right.

Necessity: \text{(Castro 2012)}
Sufficiency (uncompressed): \text{(Malloy & Nowak, 2010; Malloy & Nowak, 2011)}
Sufficiency (compressed): \text{(JH, Baraniuk, Castro, & Nowak 2012, Malloy & Nowak 2013)}
– Beyond Simple Sparsity –

The Role of Structure
Our Focus: Tree Sparsity

Characteristics of tree structure:

- Elements of $x$ in one-to-one correspondence with nodes of $\mathcal{T}$
- Nonzeros of tree-sparse vector form rooted connected subtree of $\mathcal{T}$

Question: Does tree structure help in support recovery?
Conditions for Exact Support Recovery

<table>
<thead>
<tr>
<th></th>
<th>Non-structured</th>
<th>Structured</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-adaptive</td>
<td>$\mu \geq \text{const.}\sqrt{\frac{n}{R}} \log n$</td>
<td>$\mu \geq \text{const.}\sqrt{\frac{n}{R}}$</td>
</tr>
<tr>
<td>Adaptive</td>
<td>$\mu \geq \text{const.}\sqrt{\frac{n}{R}} \log k$</td>
<td>?</td>
</tr>
</tbody>
</table>

Detection of simple trail (uncompressed sensing)

# Conditions for Exact Support Recovery

The intersection of adaptivity and (tree) structure...

<table>
<thead>
<tr>
<th></th>
<th>Non-structured</th>
<th>Structured</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-adaptive</td>
<td>$\mu \geq \text{const.}\sqrt{\left(\frac{n}{R}\right)\log n}$</td>
<td>$\mu \geq \text{const.}\sqrt{\left(\frac{n}{R}\right)}$</td>
<td>Non-adaptive</td>
</tr>
<tr>
<td>Adaptive</td>
<td>$\mu \geq \text{const.}\sqrt{\left(\frac{n}{R}\right)\log k}$</td>
<td>$\mu \geq \text{const.}\sqrt{\left(\frac{k}{R}\right)\log k}$</td>
<td>Adaptive</td>
</tr>
</tbody>
</table>

Recent related work:
Adaptive Tree Sensing: An Example

If the hypothesis test is correct at each step, then
\[ m = dk + 1 = O(k) \]

Adaptive Wavelet “Tree Sensing” in the Literature:

- Non-Fourier encoded MRI  \((\text{Panych & Jolesz, 1994})\)
- Compressive Imaging  \((\text{Deutsch, Averbuch, & Dekel, 2009})\)

(none analyzed the case of *noisy* measurements...)

\[ y(1) = x_1 + w(1) \]
\[ y(2) = x_2 + w(2) \]
\[ y(3) = x_5 + w(3) \]
\[ y(4) = x_6 + w(4) \]
\[ y(5) = x_7 + w(5) \]
Orthogonal Dictionaries and Tree Sparsity

Consider signals $z \in \mathbb{R}^p$ that are sparse in a known dictionary $D \in \mathbb{R}^{p \times n}$. That is, $z = Dx$, where

- $x \in \mathbb{R}^n$ is $k$-sparse,
- $D$ satisfies $D^T D = I_{n \times n}$, and
- columns of $D$ are $d_j, j = 1, 2, \ldots, n$

We are interested in the case where $x$ is tree-sparse...

Collect (noisy) observations of $z$ by projecting onto (scaled) columns of $D$. Suppose, for example, that the $j$-th measurement is obtained by projecting onto column $d_i$, then

$$y(j) = \beta d_i^T z + w(j)$$

where $w(j) \sim \mathcal{N}(0, 1)$.

Nonnegative scaling factor (equivalently, could consider non-unit noise variance)
Support Recovery via Adaptive Tree Sensing

**Theorem** (A. Soni & JH, 2011)
Let $T_{n,d}$ be a balanced, rooted connected tree of degree $d$ with $n$ nodes. Suppose that $z \in \mathbb{R}^p$ can be expressed as $z = Dx$, where $D$ is a known dictionary with orthonormal columns and $x$ is $k$-sparse. Further, suppose the support of $x$ corresponds to a rooted connected subtree of $T_{n,d}$. Observations of $z$ are of the form of projections of $z$ onto columns of $D$.

Let the index corresponding to the root of $T_{n,d}$ be known, and apply the top-down tree sensing procedure with threshold $\tau$ and scaling parameter $\beta$. For any $c_1 > 0$ and $c_2 \in (0, 1)$, there exists a constant $c_3 > 0$ such that if

$$\mu = \min_{i \in S} |x_i| \geq \sqrt{c_3 \beta^{-2} \log k}$$

and $\tau = c_2 \mu \beta$, the tree sensing procedure collects $m = dk + 1$ measurements, and produces a support estimate $\hat{S}$ that equals $S$ with probability at least $1 - k^{-c_1}$.

Choose $\beta = \sqrt{\frac{R}{(d+1)k}}$, then the theorem guarantees exact support recovery (whp) when

$$\mu \geq \sqrt{c_3 (d + 1) \left( \frac{k}{R} \right) \log k}$$
## Conditions for Exact Support Recovery

The intersection of adaptivity and (tree) structure...

<table>
<thead>
<tr>
<th></th>
<th>Non-structured</th>
<th>Structured</th>
<th>Non-adaptive</th>
<th>Adaptive</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-adaptive</td>
<td>$\mu \geq \text{const.} \sqrt{\left(\frac{n}{R}\right) \log n}$</td>
<td>$\mu \geq \text{const.} \sqrt{\frac{n}{R}}$</td>
<td>(S) conjecture for support recovery</td>
<td></td>
</tr>
<tr>
<td>Adaptive</td>
<td>$\mu \geq \text{const.} \sqrt{\left(\frac{n}{R}\right) \log k}$</td>
<td>$\mu \geq \text{const.} \sqrt{\left(\frac{k}{R}\right) \log k}$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(A. Soni & JH, 2011)

Akshay Soni
University of Minnesota
— LASer —

Learning Adaptive Sensing Representations
Beyond Wavelet Trees: \textit{Learned} Representations

Given training data $Z \in \mathbb{R}^{p \times q}$, want to learn a dictionary $D$ so that

$$Z \approx DX, \ D \in \mathbb{R}^{p \times n}, \ X \in \mathbb{R}^{n \times q},$$

and each column of $X$, $x_i \in \mathbb{R}^n$, is tree-sparse in $\mathcal{T}_{n,d}$.

Pose this as an optimization:

$$\{D, X\} = \arg \min_{D \in \mathbb{R}^{p \times n}, D^T D = I_{n \times n}, \ \{x_i\}} \sum_{i=1}^{q} \|z_i - Dx_i\|_2^2 + \lambda \Omega(x_i)$$

The regularization term is $\Omega(x_i) = \sum_{g \in G} \omega_g \|(x_i)_g\|$, where

- $G$ denotes a set of (overlapping) groups of indices for $x$,
- $(x_i)_g$ is $x_i$ restricted to the indices in the group $g \in G$,
- $\omega_g$ are non-negative weights, and
- the norm can be, eg., $\ell_2$ or $\ell_\infty$

Solve by \textit{alternating minimization} over $D$ and $X$  \cite{Jenatton2010}

\textbf{Sparse Modeling Software (SPAMS): http://spams-devel.gforge.inria.fr/}
Group Specifications to Enforce Tree Structure

Example: Binary Tree, 15 nodes, 4 levels...
— LASeR —

An Illustrative Example
Learning Adaptive Sensing Representations

LASeR: Learning Adaptive Sensing Representations

Learn representation for 163 images from Psychological Image Collection at Stirling (PICS) http://pics.psych.stir.ac.uk/

Example images (128 × 128)
Learned Orthogonal *Tree-Basis* Elements

(First four levels of 7 total)
Qualitative Results

“Sensing Energy”
$R = 128 \times 128$

Wavelet Tree Sensing

PCA

CS LASSO

CS Tree LASSO

LASeR

m = 20

m = 50

m = 80

original image
Qualitative Results

“Sensing Energy”

\[
R = \frac{128 \times 128}{32}
\]

Wavelet Tree Sensing

PCA

CS LASSO

CS Tree LASSO

LASeR

original image
Quantitative Results

\[
\sigma = 1
\]

\[
R = (128 \times 128)
\]

\[
R = (128 \times 128)/32
\]

\[
R = (128 \times 128)/128
\]

---: LASer

\[\square:\] PCA

\[\diamond:\] CS Lasso

\[\circ:\] CS Tree Lasso

\[\diamondsuit:\] Wavelet Sensing

\[
SNR = 10 \log_{10} \left( \frac{\|x\|_2^2}{\|x - \hat{x}\|_2^2} \right).
\]

original image
— LASeR —

Imaging via “Patch-wise” Sensing
“Patch-wise” Sensing Experiment

Motivated by EO Imaging Application (Thanks: Bob Muise @ Lockheed Martin)

Training Data:
3 Sample images from the Columbus Large Image Format (CLIF) 2007 Dataset
Each image is 1024x1024

Randomly extracted 3000 32x32 patches (at random locations)...
and vectorized them into length 1024 vectors

Applied PCA and LASeR (7-level 127 node binary tree) to this training data
In the tree-sensing context, can view PCA sensing approach in terms of a tree of degree 1
Learned Orthogonal *Tree-Basis* Elements
Example: Approximation by “Patch-wise” Sensing

Test Image (another image from CLIF database)

Sense & reconstruct non-overlapping 32x32 patches…
…comparing LASeR, PCA, Wavelets…
Sampling rate: 12.5%

Approximation Results – Uniform Sampling Rate

LASeR
$rSNR = 16.5 \text{ dB}$

PCA
$rSNR = 17.6 \text{ dB}$

$rSNR \triangleq -20 \log_{10}(\|\hat{x} - x\|_F/\|x\|_F)$
Approximation Results – Uniform Sampling Rate

Sampling rate: 12.5%

\[ \text{LASeR} \]
\[ r\text{SNR} = 16.5 \text{ dB} \]

\[ r\text{SNR} \triangleq -20 \log_{10} \left( \frac{\| \hat{x} - x \|_F}{\| x \|_F} \right) \]
Sampling rate: 12.5%

Approximation Results – Uniform Sampling Rate

PCA
rSNR = 17.6 dB

\[ rSNR \triangleq -20 \log_{10}(\|\hat{x} - x\|_F / \|x\|_F) \]
Approximation Results – Uniform Sampling Rate

Sampling rate: 12.5%

LASeR
rSNR = 16.5 dB

2D Haar Wavelet
rSNR = 13.5 dB

\[ rSNR \triangleq -20 \log_{10}(\|\hat{x} - x\|_F/\|x\|_F) \]
Approximation Results – Adaptive Sampling Rate

Average sampling rate: 7.2%

LASeR
rSNR = 13.9 dB

PCA
rSNR = 15.0 dB

\[
\text{rSNR} \triangleq -20 \log_{10}(\|\hat{x} - x\|_F/\|x\|_F)
\]
**Approximation Results – Adaptive Sampling Rate**

*Average sampling rate: 7.2%*

---

- **LASeR**
  - rSNR = 13.9 dB

- **2D Haar Wavelet**
  - rSNR = 11.9 dB

\[
\text{rSNR} \triangleq -20 \log_{10}(\| \hat{x} - x \|_F / \| x \|_F)
\]
Approximation: Zoomed In

A Closer Look... (Average sampling rate: 7.2%)
Approximation: Zoomed In

A Closer Look...  (Average sampling rate: 7.2%)
Sampling Rate Adapts to Block “Complexity”

Sampling rate per block (Average sampling rate: 7.2%)
Sampling Rate Adapts to Block “Complexity”

Sampling rate per block (Average sampling rate: 7.2%)
### Summary: Adaptivity + Structure

#### In Theory (Support Recovery)

<table>
<thead>
<tr>
<th></th>
<th>Non-structured</th>
<th>Structured</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Non-adaptive</strong></td>
<td>$\mu \geq \text{const.}\sqrt{\left(\frac{n}{R}\right) \log n}$</td>
<td>$\mu \geq \text{const.}\sqrt{\left(\frac{n}{R}\right)}$ (conj.)</td>
</tr>
<tr>
<td><strong>Adaptive</strong></td>
<td>$\mu \geq \text{const.}\sqrt{\left(\frac{n}{R}\right) \log k}$</td>
<td>$\mu \geq \text{const.}\sqrt{\left(\frac{k}{R}\right) \log k}$</td>
</tr>
</tbody>
</table>

#### Conclusions:

Polynomial reduction in SNR required for exact support recovery (for fixed “sensing energy”)
Summary: Adaptivity + Structure

In Practice (Learned Representations and Patch-wise Sensing)

Conclusions:
PCA works very well on “small” patch sizes (shared, elemental structure) in noise free settings.
Summary: Adaptivity + Structure

Conclusions:
Potential benefit for learned representations depend on patch size, data “regularity”, noise. Use LASeR on PCA residuals? (ie, fuse PCA and LASeR)

Thank You!
www.ece.umn.edu/~jdhaupt
jdhaupt@umn.edu