Compressed Sensing Applications in Microscopy and Microanalysis

Joshua Taillon
CS-Bio-Met meeting - May 18, 2017
Outline

• Introduction (about me)
• Brief (and basic) introduction to CS
• Existing applications of CS in microscopy
• Our work (in progress)
• Future ideas
Introduction
Introduction (About Me)

• NRC Postdoc in Materials Measurements Science Division
  • Microscopy and Microanalysis Research Group

• Background in Materials Characterization:
  • TEM, FIB/SEM, EELS/EDS spectroscopies
Introduction (About Me)

• NRC Postdoc in Materials Measurements Science Division
  • Microscopy and Microanalysis Research Group

• Background in Materials Characterization:
  • TEM, FIB/SEM, EELS/EDS spectroscopies

  HyperSpy
  multi-dimensional data analysis
Introduction (About Me)

• NRC Postdoc in Materials Measurements Science Division
  • Microscopy and Microanalysis Research Group

• Background in Materials Characterization:
  • TEM, FIB/SEM, EELS/EDS spectroscopies

Three dimensional nanotomography and microstructure analysis:
Introduction (About Me)

• NRC Postdoc in Materials Measurements Science Division
  • Microscopy and Microanalysis Research Group

• Background in Materials Characterization:
  • TEM, FIB/SEM, EELS/EDS spectroscopies
  • TCL
  • Matlab
  • python
Research Interests

• Lots of opportunity at intersection of microscopy and computer science/mathematics
  • Active learning
  • Novel data analysis methods
  • Automated tool control
  • Compressed sensing
Brief intro to compressed sensing
Motivating example

• Assume you have 7 coins
  • One is counterfeit with a different mass than the others
  • Easy solution would be to measure the mass of each individual coin
  • Can we do better?

Motivating example

• Assume you have 7 coins
  • One is counterfeit with a different mass than the others
  • Easy solution would be to measure the mass of each individual coin
  • Can we do better?

• Measure groupings of coins instead:
  • Only 3 measurements needed

\[ \Phi = \begin{bmatrix}
1 & 0 & 1 & 0 & 1 & 0 & 1 \\
0 & 1 & 1 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 1 & 1 & 1 & 1 \\
\end{bmatrix} \]

\( N = 7 \) coins
\( n = 3 \) measurements

Motivating example

• Assume you have 7 coins
  • One is counterfeit with a different mass than the others
  • Easy solution would be to measure the mass of each individual coin
  • Can we do better?

• Measure groupings of coins instead:
  • Only 3 measurements needed
  • More generally,
    \[ n \approx \log_2(N) \ll N \]

More general CS formalism

• \( \Phi \) is \( n \times N \) “sensing matrix”
  
  • We are trying to recover an unknown sparse vector \( \mathbf{x} \in \mathbb{R}^N \) with a measurement vector \( \mathbf{b} \) and a known sensing matrix \( \Phi \)
  
  • \( \mathbf{x} \) is sparse (mostly zeros), and describes which coin is different
  
  • We want to find a sparse solution that satisfies \( \Phi \mathbf{x} = \mathbf{b} \), given:

\[
\begin{align*}
\mathbf{b} &= \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \\
\Phi &= \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \\
\mathbf{x} &= \begin{bmatrix} ? \\ ? \\ \vdots \end{bmatrix}
\end{align*}
\]
More general CS formalism

• How do we determine $\Phi$?
  • Strictly speaking, it must obey the “restricted isometry principle” (RIP)
    • RIP means that if we select $K$ random columns from $\Phi$, that submatrix is full rank
    • Designing a $\Phi$ to satisfy the RIP is actually NP-hard, but it turns out it’s not necessary for a successful recovery of $x$, just sufficient
  
  • Turns out that a random binary matrix will satisfy RIP (with high probability)
    • In practically all implementations of CS, this random sampling is what is used
More general CS formalism

- Φ has fewer rows than columns, so Φx = b is underdetermined
  - This means there are an infinite number of solutions
- The “magic” of compressed sensing is the use of the $\ell_1$ norm for convex optimization to find the “best” x (when using the right Φ)

<table>
<thead>
<tr>
<th>Minimizing Norm</th>
<th>Definition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\ell_0$</td>
<td>$|x|_0 = \sqrt{\sum_i x_i^0} = #(i</td>
<td>x_i \neq 0)$</td>
</tr>
<tr>
<td>$\ell_1$</td>
<td>$|x|_1 = \sum_i</td>
<td>x_i</td>
</tr>
<tr>
<td>$\ell_2$</td>
<td>$|x|_2 = \sqrt{\sum_i x_i^2}$</td>
<td>Euclidean distance (Least squares difference)</td>
</tr>
</tbody>
</table>

Why “compressed” sensing?

• Consider a “compressible” HRTEM image:
  • In the pixel basis, many coefficients needed to encode the image
  • In the wavelet basis, very few coefficients are needed
  • Idea behind JPEG, etc.

• What if we could measure the sparse basis directly?

Areas of initial success - MRI

• Subsampling $k$-space greatly reduces number of acquisitions needed, and time for patient
  • Greatly expands indications for MRI (pediatrics, compromised health, etc.)
  • After 10 years, first FDA-approved applications of CS-MRI on the market in Feb. 2017 (link)

• Compressed sensing saves lives!
Areas of initial success – NMR Spectroscopy

- Multidimensional nuclear magnetic resonance
  - NMR spectra are typically sparse – few isolated peaks in the Fourier domain
  - Measured 30% of the 128 complex data pairs
  - Measurement time: 165 min → 50 min
- Observed very high fidelity reconstructions w.r.t. peak positions and shape/intensity

Efforts in CS for electron microscopy
CS in TEM (electron tomography – CS-ET)

• Tilt-tomography finds 3D representations of objects in the TEM by acquiring 2D images at many tilt angles
  • Can reduce number of tilted images needed using CS principles and compare results to the standard *simultaneous iterative reconstruction technique* (SIRT)

• Tomography of iron oxide nanoparticles:
  • CS-ET performs significantly better than SIRT, at all signal levels


CS in STEM (imaging)

- **Benefits of reduced sampling in STEM:**
  - Reduced dose (for electron-sensitive materials)
  - Reduced time/increased throughput

- **Random sampling in pixel-domain**
  - Bayesian factor analysis to find sparse representation (BPFA)
  - Sampling done with beam blanker or meandering beam

---

A. Stevens, et al., *The potential for Bayesian compressive sensing to significantly reduce electron dose in high-resolution STEM images*, Microscopy. 63, 41 (2014).
CS in SEM

• Two published implementations:
    • Use a split Bregman formulation of basis-pursuit for \( \ell_1 \) minimization
    • Drive beam to random pixels, taking care to account for scan coil dynamics
    • 10x imaging speedup when imaging only 10% of pixel locations (linear response)
  
    • Use the same BPFA as in Stevens’ CS-STEM
    • No scan coil modifications to SEM (just need high-speed beam blanker)
    • Did not demonstrate significant speed gains, but did significantly reduce dose
CS in SEM

  - Analyzed effect of reduced dose from CS-SEM on electron-sensitive human collagen sample
  - Significant reduction in sample modification due to beam-damage
- Also tested various scan patterns
  - Random sampling was found to perform better than spiral, Lissajous, and random line sampling
  - Highest reconstructed PSNR and least number of scanning artifacts
Our initial work
Our goals

- Originally set out to implement what was published by Hujsak
  - Have pivoted instead to see what impact we can have in the analytical realm
  - One conference paper on CS in STEM-EELS, but nothing else (yet)
  - Given our preference for FIB/SEM, we'll focus on X-ray analysis (EDS)
    - Long dwell times needed
    - Acquisitions will damage beam-sensitive materials
    - Seems ripe for “disruption”
Strategy

- Identify existing reconstruction algorithm that is purpose-built for hyperspectral imaging (HSI) data
  - The hyperspectral remote sensing community has been much faster to adopt advanced data analysis methods (like CS) than the microscopy community
- Demonstrate control and implement on existing HSI data
- Apply to simulated EDS data to determine/quantify effectiveness
- Determine what (if any) performance enhancement can be gained on live experiments
BPFA = Beta-Bernoulli Process Factor Analysis

- Factor analysis
  - Decompose signal into a linear approximation of factors and weights

\[ x_i = D \cdot w_i + \epsilon_i \]

- Original data samples
- Dictionary matrix of “prototypical” signals
- Dictionary element weights (vector)
- Noise and residual

An image patch \((x_i)\) can be represented by a dictionary \((D)\) of representative patches, with each element weighted by a factor from \(w\) (plus some noise \(\epsilon\))

Figure from: A. Stevens, et al. Microscopy. 63, 41 (2014).
Note: \(w_i = \alpha_i\ in this \ paper's \ notation\)
BPFA = Beta-Bernoulli Process Factor Analysis

• How to find $D$ and $w$?
  • Bayesian beta-Bernoulli Process

• We infer the underlying signal $x_i = Dw_i$ by:
  • Placing Bayesian priors on $D$, $w_i$, and $\epsilon_i$
  • Assuming that $w_i$ is sparse
  • Iterate on each parameter to improve the estimation of their values (based on observed data)

Details about algorithms in:
BPFA Formalism

• What does beta-Bernoulli Process mean in practice?
  • BP is a strategy for updating $w_i$ in a Bayesian manner

• Define $w_i$ as $z_i \odot s_i$:
  • $\odot$ = element-wise multiplication
  • $s_i$ are the dictionary weights
  • $z_i$ are “binary indicators”, specifying which of the $K$ columns of $D$ are used to represent $x_i$:
    • $z_i$ are drawn from a Bernoulli distribution:
      $$z_i \sim \prod_{k=1}^{K} \text{Bernoulli } (\pi_k)$$
    • Where $\pi_k$ is the $k$th component of:
      $$\pi \sim \prod_{k=1}^{K} \text{Beta} \left( \frac{a}{K}, \frac{b(K - 1)}{K} \right)$$

Bernoulli Distribution

$$f(k; p) \begin{cases} p & \text{if } k = 1, \\ 1 - p & \text{if } k = 0. \end{cases}$$
BPFA Formalism

• Full model parameters:
  • \( x_i = D w_i + \epsilon_i \)
  • \( w_i = z_i \odot s_i \)
  • \( d_k \sim \mathcal{N}(0, P^{-1} I_p) \)
  • \( s_i \sim \mathcal{N}(0, \gamma_s^{-1} I_K) \)
  • \( \epsilon_i \sim \mathcal{N}(0, \gamma_{\epsilon}^{-1} I_P) \)
  • \( z_i \sim \prod_{k=1}^K \text{Bernoulli}(\pi_k) \)
  • \( \pi \sim \prod_{k=1}^K \text{Beta}\left(\frac{a}{K}, \frac{b(K-1)}{K}\right) \)

• More definitions:
  • \(~\) means a Bayesian variable drawn from the specified prior distribution
  • \( \mathcal{N}(i, j) \) specifies a normal distribution with mean \( i \) and variance \( j \)
  • \( \gamma_i \) are conjugate hyperpriors of the form \( \gamma_s \sim \text{Gamma}(c, d) \) and \( \gamma_{\epsilon} \sim \text{Gamma}(e, f) \)
  • \( d_k \) represents the \( k \)th component (column) of \( D \)
How does it work?

• The model is inferred using **Gibbs Sampling**
  • Markov chain Monte Carlo algorithm used to sample the full Bayesian likelihood
  • Samples from the posterior distribution of each random variable are estimated by iteratively sampling from the conditional distributions of each variable (given all the others)
  • After “enough” iterations, the estimated variables are likely close to the true model parameters

• **Such a method determines both the dictionary and weights iteratively**
  • Can be used on plain imagery (like Anderson and Hujsak)
  • Also developed for hyperspectral imagery (by Xing)
What does it look like in practice?

- Example data from algorithm authors:
  - Hyperspectral imagery (HSI) from satellite imaging of an urban environment
  - 150 x 150 spatial pixels – 210 spectral bands
  - Artificially remove some large fraction of voxels, and reconstruct using BPFA:
What does it look like in practice?

<table>
<thead>
<tr>
<th>Original data</th>
<th>2% sampling</th>
<th>2x2 patch size</th>
<th>4x4 patch size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spectral Channel 100</td>
<td>PSNR = 8.77 dB</td>
<td>PSNR = 21.36 dB</td>
<td>PSNR = 24.00 dB</td>
</tr>
<tr>
<td>Center pixel spectrum</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
How about EDS?

- Simulated 3D data cube using DTSA-II:
  - 100 x 100 spatial – 2048 spectral channels

Schematic of spherical geometry

3D visualization of quantified block
How about EDS?

• Simulated 3D data cube using DTSA-II:
  • 100 x 100 map – 2048 spectral channels

• Fully sampled quantification data:

Carbon  Platinum  Calcium  Phosphorus  Silicon  Copper
BPFA reconstruction of subsampled EDS

- Tested both 2 x 2 and 4 x 4 patch sizes
- Subsample by zeroing out a large fraction of the simulated voxels
- Analyzed reconstruction output as function of observation ratio
Reconstruction “phase transition”

- Donoho-Tanner transition
  - Reconstruction either fails completely, or does pretty well
  - Abrupt onset of success at a certain signal level

\[
PSNR = 10 \cdot \log_{10} \frac{MAX_i^2}{MSE}
\]
Reconstruction “phase transition”

- Donoho-Tanner transition
  - Reconstruction either fails completely, or does pretty well
  - Abrupt onset of success at a certain signal level
  - Quality of reconstruction continues to improve with more data

$$PSNR = 10 \cdot \log_{10} \frac{MAX^2}{MSE}$$
More realistic test

• Random subsampling of entire datacube is not particularly realistic

• Instead, simulate EDS collections with varying dwell times
  • Simulates faster experimental maps, without subsampling locations (or energies)
Results of live time variation

Input at 2.01 keV

0.005 s  
0.010 s  
0.015 s  
0.020 s  
0.025 s  
0.030 s

Output at 2.01 keV

0.005 s  
0.010 s  
0.015 s  
0.020 s  
0.025 s  
0.030 s
Results of live time variation

• Improvement in PSNR statistics not as dramatic as true random sampling
  • Never reaches high 20s value of previous example

• Reasons?
  • Suspect not sparse enough in energy axis
  • Thresholding of continuum X-rays could help
  • Still investigating these results
Future directions
Ongoing work

• **Improving results of reduced dwell time reconstructions**
  • X-ray continuum makes signal non-sparse, leading to bad performance
  • Implement some sort of thresholding, or artificially subsample energy dimension?
  • Eventually need to demonstrate effectiveness on experimental data

• **Extension of algorithm to 3D**
  • Should be relatively simple, and could enable even lower electron doses

• **Extend algorithm to allow incoming information**
  • Could make interactive EDS map collections more immediately informative

• **Make code more performant**
  • Currently single-threaded in Matlab and has not been optimized at all
Thank you!

Questions/comments?

joshua.taillon@nist.gov
x2913