The Basics of Sparse Representations

Justin Romberg, Georgia Tech ECE
CCNS Summer School, SAMS, RTP, NC
July 27, 2015
Signal/image \( f(t) \) in the time/spatial domain

Decompose \( f \) as a *superposition of atoms*

\[
f(t) = \sum_i \alpha_i \psi_i(t)
\]

\( \psi_i = \) basis functions

\( \alpha_i = \) expansion coefficients in \( \psi \)-domain

Classical example: *Fourier series*

\( \psi_i = \) complex sinusoids

\( \alpha_i = \) Fourier coefficients

Modern example: *wavelets*

\( \psi_i = \) “little waves”

\( \alpha_i = \) wavelet coefficients
Taking images apart and putting them back together

- Frame operators $\Psi$, $\tilde{\Psi}$ map images to sequences and back
- Two sequences of functions: $\{\psi_i(t)\}, \{\tilde{\psi}(t)\}$
- Analysis (inner products):
  \[ \alpha = \tilde{\Psi}^*[f], \quad \alpha_i = \langle \tilde{\psi_i}, f \rangle \]
- Synthesis (superposition):
  \[ f = \Psi[\alpha], \quad f = \sum_i \alpha_i \psi_i(t) \]
- If $\{\psi_i(t)\}$ is an orthobasis, then
  \[ \|\alpha\|_{\ell^2}^2 = \|f\|_{L^2}^2 \]  (Parseval)
  \[ \sum_i \alpha_i \beta_i = \int f(t) g(t) \, dt \]  (where $\beta = \tilde{\Psi}[g]$)
  \[ \psi_i(t) = \tilde{\psi_i}(t) \]
  i.e. all sizes and angles are preserved
- Overcomplete tight frames have similar properties
ACHA Mission: construct “good representations” for “signals/images” of interest

Examples of “signals/images” of interest
- Classical: signal/image is “bandlimited” or “low-pass”
- Modern: smooth between isolated singularities (e.g. 1D piecewise poly)
- Postmodern: 2D image is smooth between smooth edge contours

Properties of “good representations”
- sparsifies signals/images of interest
- can be computed using fast algorithms ($O(N)$ or $O(N \log N)$ — think of the FFT)
Example: The discrete cosine transform (DCT)

- For an image \( f(t, s) \) on \([0, 1]^2\), we have

\[
\psi_{\ell,m}(t, s) = 2\lambda_\ell \lambda_m \cdot \cos(\pi \ell t) \cos(\pi ms), \quad \lambda_\ell = \begin{cases} 
1/\sqrt{2} & \ell = 0 \\
1 & \text{otherwise}
\end{cases}
\]

- Closely related to 2D Fourier series/DFT, the DCT is real, and implicitly does symmetric extension
- Can be taken on the whole image, or blockwise (JPEG)
Image approximation using DCT

Take 1% of “low pass” coefficients, set the rest to zero

original

approximated

rel. error = 0.075
Image approximation using DCT

Take 1\% of “low pass” coefficients, set the rest to zero

original

approximated

rel. error = 0.075
Image approximation using DCT

Take 1% of largest coefficients, set the rest to zero (adaptive)

original

approximated

rel. error = 0.057
Image approximation using DCT

Take 1% of *largest* coefficients, set the rest to zero (adaptive)

original

approximated

rel. error = 0.057
Wavelets

\[ f(t) = \sum_{j,k} \alpha_{j,k} \psi_{j,k}(t) \]

- **Multiscale:** indexed by scale \( j \) and location \( k \)
- **Local:** \( \psi_{j,k} \) analyzes/represents an interval of size \( \sim 2^{-j} \)
- **Vanishing moments:** in regions where \( f \) is polynomial, \( \alpha_{j,k} = 0 \)
2D wavelet transform

- Sparse: few large coeffs, many small coeffs
- Important wavelets cluster along edges
Multiscale approximations

Scale = 4, 16384:1

rel. error = 0.29
Multiscale approximations

Scale = 5, 4096:1

rel. error = 0.22
Multiscale approximations

Scale = 6, 1024:1

rel. error = 0.16
Multiscale approximations

Scale = 7, 256:1

rel. error = 0.12
Multiscale approximations

Scale = 8, 64:1

rel. error = 0.07
Multiscale approximations

Scale = 9, 16:1

rel. error = 0.04
Multiscale approximations

Scale = 10, 4:1

rel. error = 0.02
Image approximation using wavelets

Take 1% of *largest* coefficients, set the rest to zero (adaptive)

original  
approximated

rel. error = 0.031
DCT/wavelets comparison

Take 1% of \textit{largest} coefficients, set the rest to zero (adaptive)

DCT

rel. error = 0.057

wavelets

rel. error = 0.031
Linear approximation

- Linear $S$-term approximation: keep $S$ coefficients in **fixed locations**

$$f_S(t) = \sum_{m=1}^{S} \alpha_m \psi_m(t)$$

- projection onto fixed subspace
- lowpass filtering, principle components, etc.

- **Fast coefficient decay** $\Rightarrow$ **good approximation**

$$|\alpha_m| \lesssim m^{-r} \quad \Rightarrow \quad \|f - f_S\|_2^2 \lesssim S^{-2r+1}$$

- Take $f(t)$ periodic, $d$-times continuously differentiable, $\Psi$ = Fourier series:

$$\|f - f_S\|_2^2 \lesssim S^{-2d}$$

*The smoother the function, the better the approximation*

Something similar is true for wavelets ...
Nonlinear approximation

Nonlinear $S$-term approximation: keep $S$ largest coefficients

$$f_S(t) = \sum_{\gamma \in \Gamma_S} \alpha_\gamma \psi_\gamma(t), \quad \Gamma_S = \text{locations of } S \text{ largest } |\alpha_m|$$

Fast decay of sorted coefficients $\Rightarrow$ good approximation

$$|\alpha|_{(m)} \lesssim m^{-r} \quad \Rightarrow \quad \|f - f_S\|_2^2 \lesssim S^{-2r+1}$$

$|\alpha|_{(m)} = m\text{th largest coefficient}$
## Linear v. nonlinear approximation

- For $f(t)$ *uniformly smooth* with $d$ “derivatives”

  \[
  S\text{-term approx. error} = S^{-2d+1}
  \]

<table>
<thead>
<tr>
<th></th>
<th>$S^{-2d+1}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fourier, linear</td>
<td></td>
</tr>
<tr>
<td>Fourier, nonlinear</td>
<td></td>
</tr>
<tr>
<td>wavelets, linear</td>
<td></td>
</tr>
<tr>
<td>wavelets, nonlinear</td>
<td></td>
</tr>
</tbody>
</table>

- For $f(t)$ *piecewise smooth*

  \[
  S\text{-term approx. error} = S^{-1}
  \]

<table>
<thead>
<tr>
<th></th>
<th>$S^{-1}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fourier, linear</td>
<td></td>
</tr>
<tr>
<td>Fourier, nonlinear</td>
<td></td>
</tr>
<tr>
<td>wavelets, linear</td>
<td></td>
</tr>
<tr>
<td>wavelets, nonlinear</td>
<td>$S^{-2d+1}$</td>
</tr>
</tbody>
</table>

Nonlinear wavelet approximations *adapt* to singularities.
Wavelet adaptation

piecewise polynomial $f(t)$

wavelet coeffs $\alpha_{j,k}$
Approximation curves

Approximating Pau with $S$-terms...

wavelet nonlinear, DCT nonlinear, DCT linear
Approximation comparison

original

DCT linear (.075)

DCT nonlinear (.057)

wavelet nonlinear (.031)
The ACHA paradigm

Sparse representations yield algorithms for (among other things)

1. compression,
2. estimation in the presence of noise ("denoising"),
3. inverse problems (e.g. tomography),
4. acquisition (compressed sensing)

that are

- fast,
- relatively simple,
- and produce (nearly) optimal results

We have looked at some *fixed* representations, it is also possible to *learn* the sparse basis.