An Overview of Sparsity with Applications to Compression, Restoration, and Inverse Problems

Justin Romberg

Georgia Tech, School of ECE

ENS Winter School
January 9, 2012
Lyon, France

Applied and Computational Harmonic Analysis

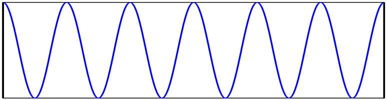
- Signal/image $f(t)$ in the time/spatial domain
- Decompose f as a *superposition of atoms*

$$f(t) = \sum_i \alpha_i \psi_i(t)$$

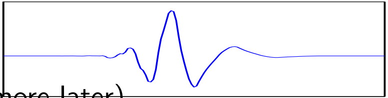
$\psi_i =$ basis functions

$\alpha_i =$ expansion coefficients in ψ-domain

- Classical example: Fourier series
  $\psi_i =$ complex sinusoids
  $\alpha_i =$ Fourier coefficients



- Modern example: wavelets
  $\psi_i =$ "little waves"
  $\alpha_i =$ wavelet coefficients



- More exotic example: curvelets (more later)

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):
  $$\alpha = \tilde{\Psi}[f], \qquad \alpha_i = \langle \tilde{\psi}_i, f \rangle$$
  Synthesis (superposition):
  $$f = \Psi^*[\alpha], \qquad f = \sum_i \alpha_i \psi_i(t)$$
- If $\{\psi_i(t)\}$ is an orthobasis, then
  $$\|\alpha\|_{\ell_2}^2 = \|f\|_{L_2}^2 \qquad \text{(Parseval)}$$
  $$\sum_i \alpha_i \beta_i = \int f(t) g(t)\, dt \qquad \text{(where } \beta = \tilde{\Psi}[g])$$
  $$\psi_i(t) = \tilde{\psi}_i(t)$$
  i.e. all sizes and angles are preserved
- Overcomplete tight frames have similar properties

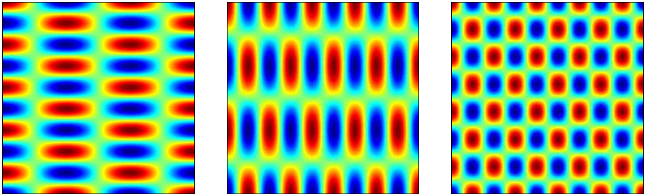
ACHA

- 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)
  - Cutting-edge: 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

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



- 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

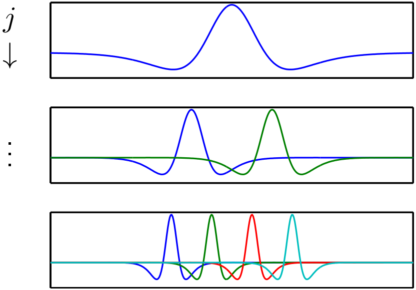
rel. error = 0.057

Wavelets

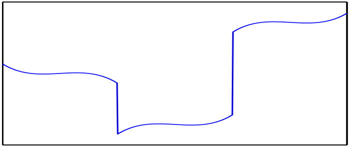
$$f(t) = \sum_{j,k} \alpha_{j,k} \psi_{j,k}(t)$$

- 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, $\alpha_{j,k} = 0$

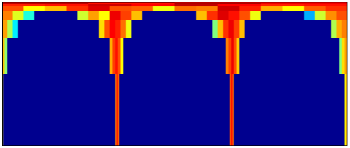
$\psi_{j,k}$



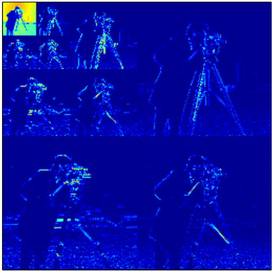
piecewise poly f

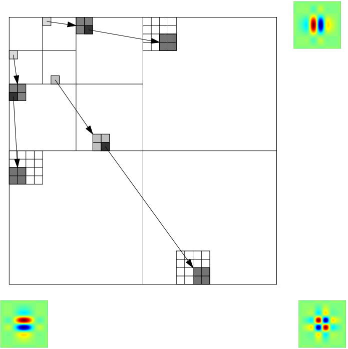


wavelet coeffs $\alpha_{j,k}$



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

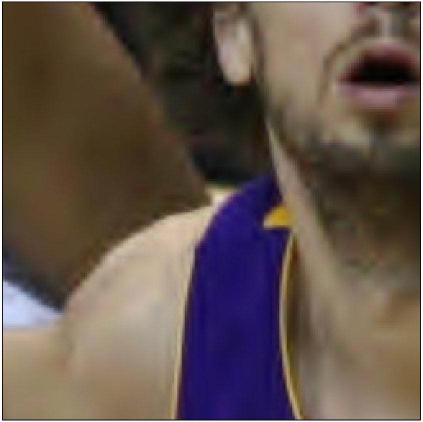
rel. error = 0.02

Image approximation using wavelets

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

original

approximated



rel. error = 0.031

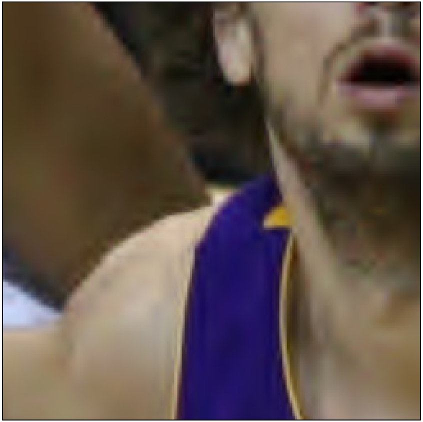
DCT/wavelets comparison

Take 1% of *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

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

- Fast decay of sorted coefficients $\Rightarrow$ good approximation

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

$|\alpha|_{(m)} = m$th largest coefficient

Linear v. nonlinear approximation

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

	S-term approx. error
Fourier, linear	S^{-2d+1}
Fourier, nonlinear	S^{-2d+1}
wavelets, linear	S^{-2d+1}
wavelets, nonlinear	S^{-2d+1}

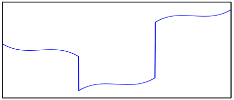
- For $f(t)$ *piecewise smooth*

	S-term approx. error
Fourier, linear	S^{-1}
Fourier, nonlinear	S^{-1}
wavelets, linear	S^{-1}
wavelets, nonlinear	S^{-2d+1}

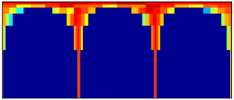
Nonlinear wavelet approximations *adapt* to singularities

Wavelet adaptation

piecewise polynomial $f(t)$

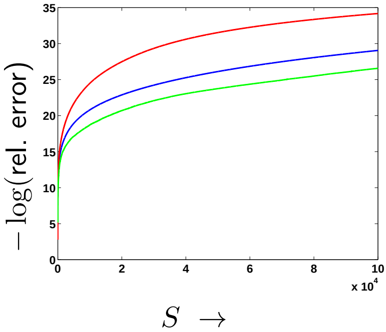


wavelet coeffs $\alpha_{j,k}$



Approximation curves

Approximating Pau with S-terms...



wavelet nonlinear, DCT nonlinear, DCT linear

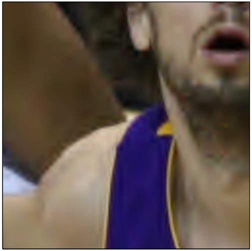
Approximation comparison

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

Compression

Transform-domain image coding

- Sparse representation = good compression
  Why? Because there are fewer things to code

- Basic, "stylized" image coder
  1. Transform image into sparse basis
  2. Quantize
     Most of the xform coefficients are ≈ 0
     $\Rightarrow$ they require very few bits to encode
  3. Decoder: simply apply inverse transform to quantized coeffs

Image compression

- Classical example: JPEG (1980s)
  - standard implemented on every digital camera
  - representation $=$ Local Fourier
    discrete cosine transform on each 8×8 block
- Modern example: JPEG2000 (1990s)
  - representation $=$ wavelets
    Wavelets are much sparser for images with edges
  - about a factor of 2 better than JPEG in practice
    half the space for the same quality image

JPEG vs. JPEG2000

Visual comparison at 0.25 bits per pixel ($\approx$ 100:1 compression)

JPEG

JPEG2000

(Images from David Taubman, University of New South Wales)

Sparse transform coding is asymptotically optimal

Donoho, Cohen, Daubechies, DeVore, Vetterli, and others ...

- The statement "transform coding in a sparse basis is a smart thing to do" can be made mathematically precise
- Class of images $\mathcal{C}$
- Representation $\{\psi_i\}$ (orthobasis) such that

$$|\alpha|_{(n)} \lesssim n^{-r}$$

  for all $f \in \mathcal{C}$ ($|\alpha|_{(n)}$ is the nth largest transform coefficient)
- Simple transform coding: transform, quantize (throwing most coeffs away)
- $\ell(\epsilon)$ = length of code (# bits) that guarantees the error $< \epsilon$ for all $f \in \mathcal{C}$ (worst case)
- To within $\log$ factors

$$\ell(\epsilon) \asymp \epsilon^{-1/\gamma}, \qquad \gamma = r - 1/2$$

- For piecewise smooth signals and $\{\psi_i\}$ = wavelets,
  no coder can do fundamentally better

Statistical Estimation

Statistical estimation setup

$$y(t) = f(t) + \sigma z(t)$$

- y: data
- f: object we wish to recover
- z: stochastic error; assume z_t i.i.d. $N(0, 1)$
- σ: noise level
- The quality of an estimate $\tilde{f}$ is given by its risk (expected mean-square-error)

$$\text{MSE}(\tilde{f}, f) = E\|\tilde{f} - f\|_2^2$$

Transform domain model

$$y = f + \sigma z$$

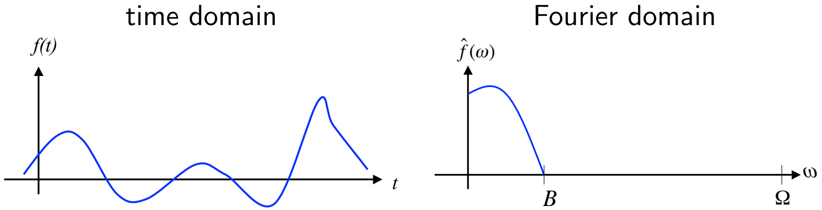
Orthobasis $\{\psi_i\}$:

$$\begin{array}{ccccc} \langle y, \psi_i \rangle & = & \langle f, \psi_i \rangle & + & \langle z, \psi_i \rangle \\ \tilde{y}_i & = & \alpha_i & + & z_i \end{array}$$

- z_i Gaussian white noise sequence
- σ noise level
- $\alpha_i = \langle f, \psi_i \rangle$ coordinates of f

Classical estimation example

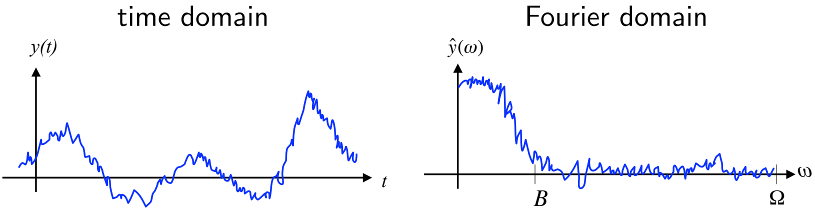
- Classical model: signal of interest f is lowpass



- Observable frequencies: $0 \leq \omega \leq \Omega$
- $\hat{f}(\omega)$ is nonzero only for $\omega \leq B$

Classical estimation example

- Add noise: $y = f + z$

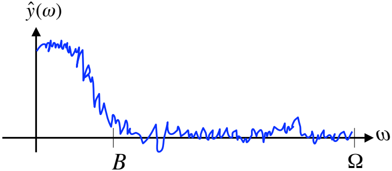


Observation error: $E\|y - f\|_2^2 = E\|\hat{y} - \hat{f}\|_2^2 = \Omega \cdot \sigma^2$

- Noise is spread out over entire spectrum

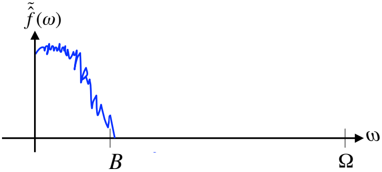
Classical estimation example

- Optimal recovery algorithm: lowpass filter (“kill” all $\hat{y}(\omega)$ for $\omega > B$)



Original error

$$E\|\hat{y} - \hat{f}\|_2^2 = \Omega \cdot \sigma^2$$



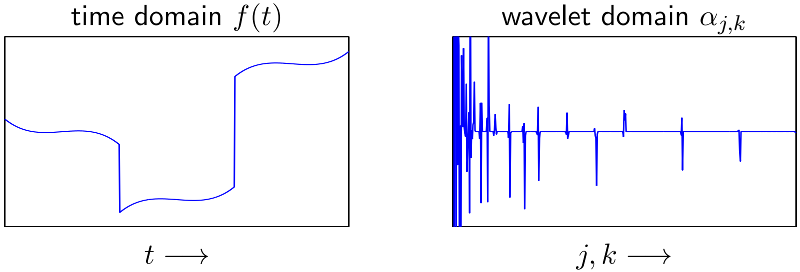
Recovered error

$$E\|\tilde{\hat{f}} - \hat{f}\|_2^2 = B \cdot \sigma^2$$

- Only the lowpass noise affects the estimate, a savings of $(B/\Omega)^2$

Modern estimation example

- Model: signal is piecewise smooth
- Signal is sparse in the wavelet domain



- Again, the $\alpha_{j,k}$ are concentrated on a small set
- This set is signal dependent (and unknown a priori)
  $\Rightarrow$ we don't know where to "filter"

Ideal estimation

$$y_i = \alpha_i + \sigma z_i, \qquad y \sim \mathrm{Normal}(\alpha, \sigma^2 I)$$

- Suppose an "oracle" tells us which coefficients are above the noise level
- Form the oracle estimate

$$\tilde{\alpha}_i^{\mathrm{orc}} = \begin{cases} y_i, & \text{if} \quad |\alpha_i| > \sigma \\ 0, & \text{if} \quad |\alpha_i| \leq \sigma \end{cases}$$

keep the observed coefficients above the noise level, ignore the rest

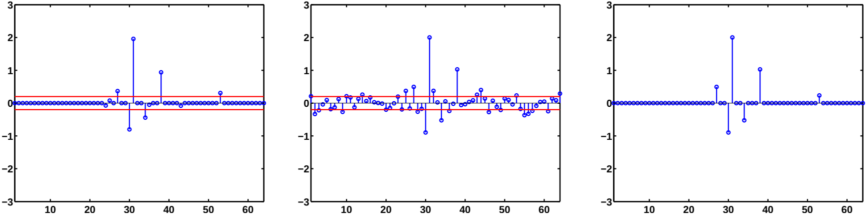
- Oracle Risk:

$$E\|\tilde{\alpha}_i^{\mathrm{orc}} - \alpha\|_2^2 = \sum_i \min(\alpha_i^2, \sigma^2)$$

Ideal estimation

- Transform coefficients α
  - Total length $N = 64$
  - $\#$ nonzero components $= 10$
  - $\#$ components above the noise level $S = 6$

original coeffs α | noisy coeffs y | oracle estimate $\tilde{\alpha}_{\mathrm{orc}}$



$$E\|y - \alpha\|_2^2 = N \cdot \sigma^2 \qquad E\|\tilde{\alpha}_{\mathrm{orc}} - f\|_2^2 = S \cdot \sigma^2$$

Interpretation

$$\text{MSE}(\tilde{\alpha}^{\text{orc}}, \alpha) = \sum_i \min(\alpha_i^2, \sigma^2)$$

- Rearrange the coefficients in decreasing order $|\alpha|^2_{(1)} \geq |\alpha|^2_{(2)} \geq \ldots \geq |\alpha|^2_{(N)}$
- S: number of those α_i's s.t. $\alpha_i^2 \geq \sigma^2$

$$\begin{aligned} MSE(\tilde{\alpha}^{\text{orc}}, \alpha) &= \sum_{i>S} |\alpha|^2_{(i)} \; + \; S \cdot \sigma^2 \\ &= \|\alpha - \alpha_S\|_2^2 \; + \; S \cdot \sigma^2 \\ &= \text{ Approx Error } + \text{ Number of terms} \times \text{noise level} \\ &= \textit{Bias}^2 \; + \; \textit{Variance} \end{aligned}$$

- The sparser the signal,
  - the better the approximation error (lower bias), and
  - the fewer # terms above the noise level (lower variance)
- *Can we estimate as well without the oracle?*

Denoising by thresholding

- Hard-thresholding ("keep or kill")

$$\tilde{\alpha}_i = \begin{cases} y_i, & |y_i| \geq \lambda \\ 0, & |y_i| < \lambda \end{cases}$$

- Soft-thresholding ("shrinkage")

$$\tilde{\alpha}_i = \begin{cases} y_i - \lambda, & y_i \geq \lambda \\ 0, & -\lambda < y_i < \lambda \\ y_i + \lambda, & y_i \leq -\lambda \end{cases}$$

- Take λ a little bigger than σ
- Working assumption: whatever is above λ is signal, whatever is below is noise

Denoising by thresholding

- Thresholding performs (almost) as well as the oracle estimator!
- Donoho and Johnstone:
  Form estimate $\tilde{\alpha}^t$ using threshold $\lambda = \sigma\sqrt{2\log N}$,

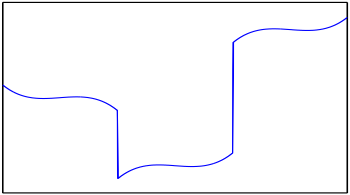
$$\mathrm{MSE}(\tilde{\alpha}^t, \alpha) := E\|\tilde{\alpha}^t - \alpha\|_2^2 \le (2\log N + 1) \cdot (\sigma^2 + \sum_i \min(\alpha_i^2, \sigma^2))$$

- Thresholding comes within a $\log$ factor of the oracle performance
- The $(2\log N + 1)$ factor is the price we pay for not knowing the locations of the important coeffs
- Thresholding is simple and effective
- Sparsity $\Rightarrow$ good estimation

Recall: Modern estimation example

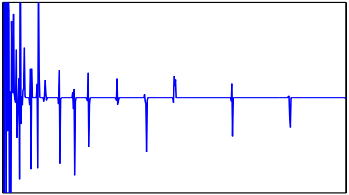
- Signal is piecewise smooth, and sparse in the wavelet domain

time domain $f(t)$



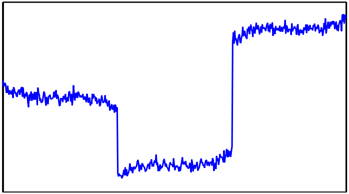
$t \longrightarrow$

wavelet domain $\alpha_{j,k}$



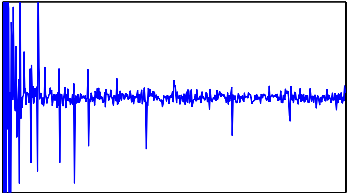
$j, k \longrightarrow$

noisy signal $y(t)$



$t \longrightarrow$

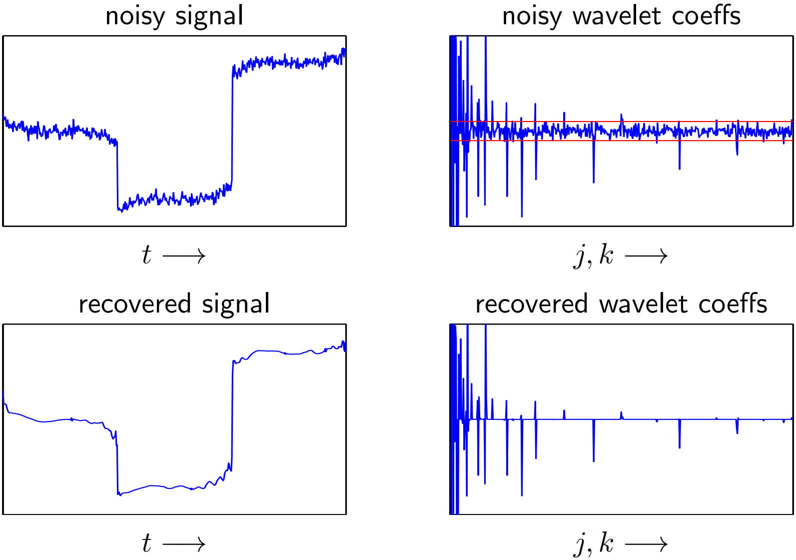
noisy wavelet coeffs



$j, k \longrightarrow$

Thresholding wavelets

- Denoise (estimate) by soft thresholding



Denoising the Phantom

noisy

Error $= 25.0$

lowpass filtered

Error $= 42.6$

wavelet thresholding, $\lambda = 3\sigma$

Error $= 11.0$

Inverse Problems

Linear inverse problems

$$y(u) = (Kf)(u) + z(u), \quad u = \text{measurement variable/index}$$

- $f(t)$ object of interest
- K linear operator, indirect measurements

$$(Kf)(u) = \int k(u,t) f(t)\, dt$$

  Examples:
  - Convolution ("blurring")
  - Radon (Tomography)
  - Abel
- $z =$ noise
- Ill-posed: $f = K^{-1} y$ not well defined

Solving inverse problems using the SVD

$$K = U\Lambda V^T$$

$$U = \mathrm{col}(u_1, \ldots, u_n), \quad \Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_n), \quad V = \mathrm{col}(v_1, \ldots, v_n)$$

- $U =$ orthobasis for the measurement space,
  $V =$ orthobasis for the signal space
- Rewrite action of operator in terms of these bases:

$$y(\nu) = (Kf)(\nu) \Leftrightarrow \langle u_\nu, y\rangle = \lambda_\nu \langle v_\nu, f\rangle$$

- The inverse operator is also natural:

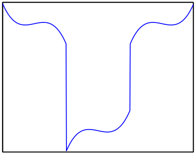
$$\langle v_\nu, f\rangle = \lambda_\nu^{-1}\langle u_\nu, y\rangle, \qquad f = V \begin{pmatrix} \lambda_1^{-1}\langle u_1, y\rangle \\ \lambda_2^{-1}\langle u_2, y\rangle \\ \vdots \end{pmatrix}$$

- But in general, $\lambda_v \to 0$, making this unstable

Deconvolution

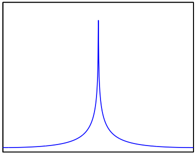
- Measure $y = Kf + \sigma z$, where K is a convolution operator

signal $f(t)$



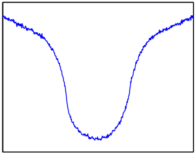
$\circledast$

convolution kernel



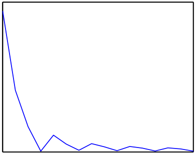
$+$ noise $=$

observed $y(t)$



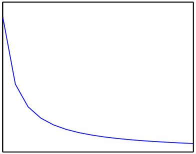
- Singular basis: $U = V =$ Fourier transform

$\{ \langle e_\nu, f \rangle \}$



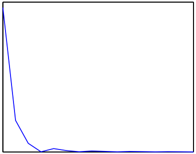
$\times$

$\{ \lambda_\nu \}$



$+$ noise $=$

$\{ \langle h_\nu, y \rangle \}$



Regularization

- Reproducing formula

$$f = \sum_\nu \lambda_\nu^{-1} \langle u_\nu, Kf \rangle v_\nu$$

- Noisy observations

$$y = Kf + \sigma z \quad \Leftrightarrow \quad \langle u_\nu, y \rangle = \langle u_\nu, Kf \rangle + \sigma \hat{z}_\nu$$

- Multiply by damping factors w_ν to reconstruct from observations y

$$\tilde{f} = \sum_\nu w_\nu \lambda_\nu^{-1} \langle u_\nu, y \rangle v_\nu$$

  want $w_\nu \approx 0$ when λ_ν^{-1} is large (to keep the noise from exploding)
- If spectral density $\theta_\nu^2 = |\langle f, v_\nu \rangle|^2$ is known, the MSE optimal weights are

$$w_\nu = \frac{\theta_\nu^2}{\theta_\nu^2 + \sigma^2} = \frac{\text{signal power}}{\text{signal power} + \text{noise power}}$$

  This is the Wiener Filter

Ideal damping

- In the SVD domain:

$$y_\nu = \theta_\nu + \sigma_\nu z_\nu$$

$y_\nu = \langle u_\nu, y \rangle, \quad \theta_\nu = \langle f, v_\nu \rangle, \quad \sigma_\nu = \sigma/\lambda_\nu, \quad z_\nu \sim$ iid Gaussian

- Again, suppose an oracle tells us which of the θ_ν are above the noise level
- Oracle "keep or kill" window (minimizes MSE)

$$w_\nu = \begin{cases} 1 & |\theta_\nu| > \sigma_\nu \\ 0 & \text{otherwise} \end{cases}$$

Take $\tilde{\theta}_\nu = w_\nu y_\nu$ (thresholding)

- Since V is an isometry, oracle risk is

$$E\|f - \tilde{f}\|_2^2 = E\|\theta - \tilde{\theta}\|_2^2 = \sum_\nu \min(\theta_\nu^2, \sigma_\nu^2)$$

Interpretation

$$
\begin{aligned}
MSE &= \sum_{\nu} \min(\theta_\nu^2, \sigma_\nu^2) \\
&= \sum_{\nu:|\theta_\nu|\lambda_\nu \le \sigma} \theta_\nu^2 + \sum_{\nu:|\theta_\nu|\lambda_\nu > \sigma} \frac{\sigma^2}{\lambda^2} \\
&= \text{Bias}^2 + \text{Variance}
\end{aligned}
$$

- Again, concentration of the $\theta_\nu := \langle f, v_\nu \rangle$ on a small set is critical for good performance
- But the v_ν are determined only by the operator K !

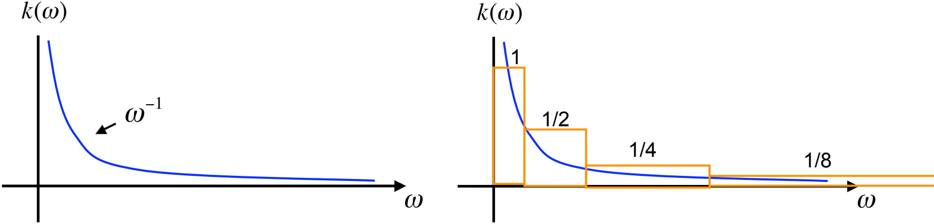
Typical Situation

- Convolutions, Radon inversion (tomography)
- $(v_\nu) \sim$ sinusoids
- f has discontinuities (earth, brain, ...)
- SVD basis is *not* a good representation for our signal

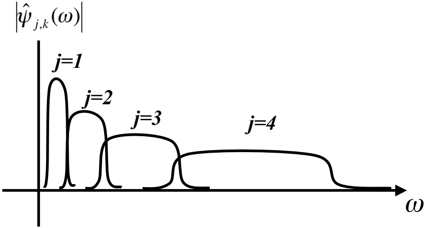
- Fortunately, we can find a representation that is simultaneously
  - almost an SVD
  - A sparse decomposition for object we are interested in

Example: Power-law convolution operators

- $K =$ convolution operator with Fourier spectrum $\sim \omega^{-1}$



- Wavelets have dyadic (in scale j) support in Fourier domain



- Spectrum of K is almost constant (within a factor of 2) over each subband

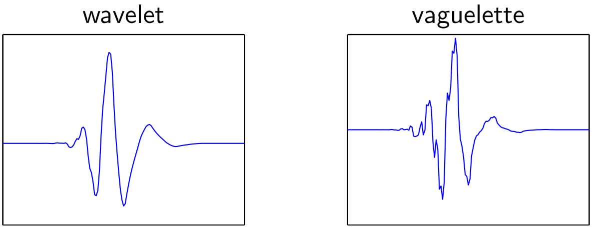
The Wavelet-Vaguelette decomposition (WVD)

Donoho, 1995

- Wavelet basis $\{\psi_{j,k}\}$ sparsifies piecewise smooth signals
- Vaguelette dual basis $u_{j,k}$ satisfies

$$\langle f, \psi_{j,k} \rangle = 2^{j/2} \langle u_{j,k}, Kf \rangle$$

  (basis for the measurement space)
- For power-law K, vaguelettes $\approx$ orthogonal, and $\approx$ wavelets



- Wavelet-Vaguelette decomposition is almost an SVD for Fourier power-law operators

Deconvolution using the WVD

- Observe $y = Kf + \sigma z$,
  $K = 1/|\omega|$ power-law operator, $z =$ iid Gaussian noise
- Expand y in vaguelette basis

$$v_{j,k} = \langle u_{j,k}, y \rangle$$

  almost orthonormal, so noise in new basis is $\approx$ independent
- Soft-threshold

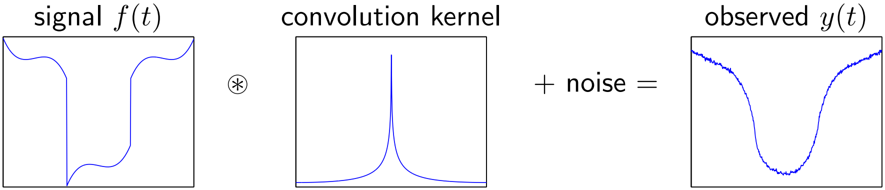
$$\tilde{v}_{j,k} = \begin{cases} v_{j,k} - \gamma \operatorname{sign}(v_{j,k}) & |v_{j,k}| > \gamma_j \\ 0 & |v_{j,k}| \leq \gamma_j \end{cases}$$

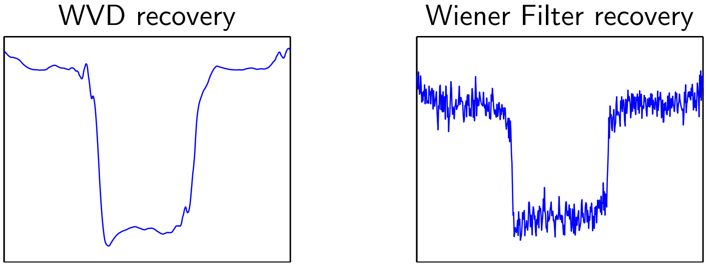
  for $\gamma_j \sim 2^{j/2}\sigma$
- Weighted reconstruction in the wavelet basis

$$\tilde{f}(t) = \sum_{j,k} 2^{j/2} \tilde{v}_{j,k} \psi_{j,k}(t)$$

Deconvolution example

- Measure $y = Kf + \sigma z$, where K is $1/|\omega|$

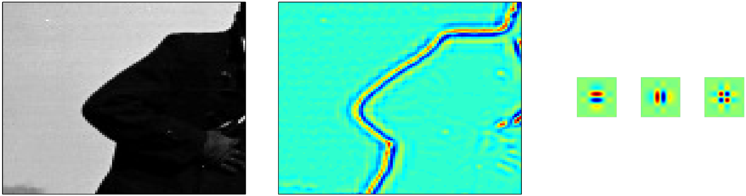




Later this week: Acquisition
(Compressed Sensing)

Curvelets

Wavelets and geometry



- Wavelet basis functions are isotropic
  $\Rightarrow$ they cannot adapt to *geometrical structure*
- Curvelets offer a more refined scaling concept...

Curvelets

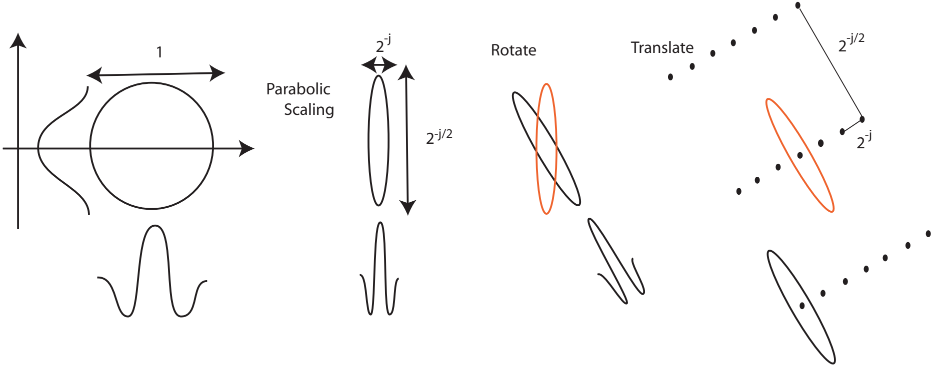
Candes and Donoho, 1999–2004

New multiscale pyramid:

- Multiscale
- Multi-orientations
- *Parabolic scaling (anisotropy)*

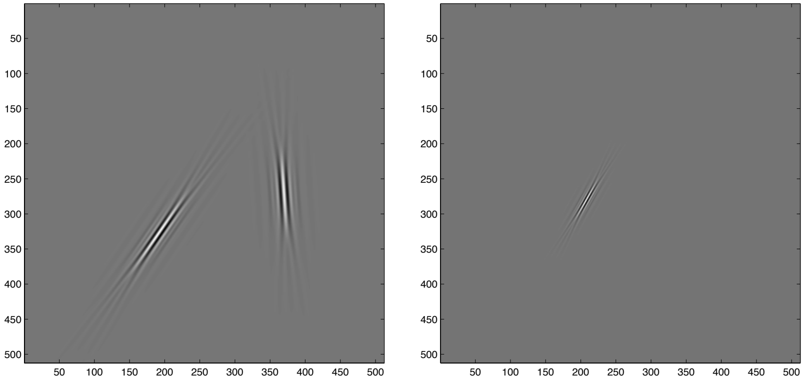
$$\text{width} \approx \text{length}^2$$

Curvelets in the spatial domain

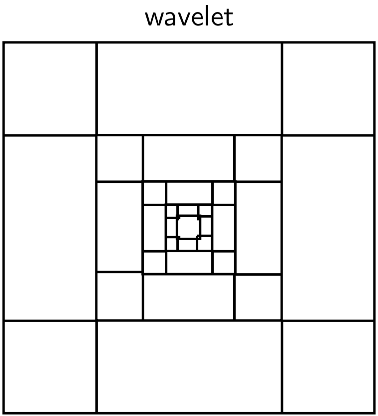


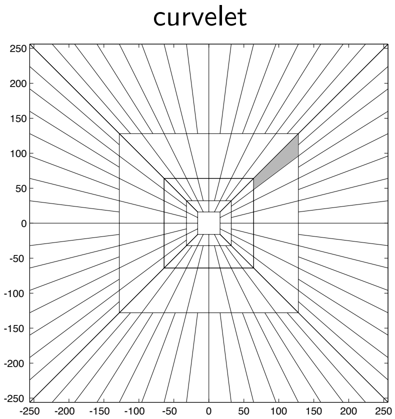
Curvelets parameterized by *scale*, *location*, and *orientation*

Example curvelets



Curvelet tiling in the frequency domain





Piecewise-smooth approximation

- Image fragment: C^2 smooth regions separated by C^2 contours
- Fourier approximation

$$\|f - f_S\|_2^2 \;\lesssim\; S^{-1/2}$$

- Wavelet approximation

$$\|f - f_S\|_2^2 \;\lesssim\; S^{-1}$$

- Curvelet approximation

$$\|f - f_S\|_2^2 \;\lesssim\; S^{-2}\log^3 S$$

(within $\log$ factor of optimal)

Application: Curvelet denoising I

Zoom-in on piece of phantom

noisy | wavelet thresholding | curvelet thresholding

Application: Curvelet denoising II

Zoom-in on piece of Lena

wavelet thresholding

curvelet thresholding

Summary

- Having a sparse representation plays a fundamental role in how well we can
  - compress
  - denoise
  - restore

  signals and images
- The above were accomplished with relatively simple algorithms (in practice, we use similar ideas + a bag a tricks)
- Better representation (e.g. curvelets) $\longrightarrow$ better results

- Wednesday and Friday:
  We will see how sparsity can play a role in *data acquisition*