Today:
- Signal recovery via sparsity
- Overview of compressed sensing
- Start two-channel filter banks—orthogonal case

Readings:
- Chapter 6 of *The World of Fourier and Wavelets*
- Full book posted Monday evening
  - Updates will be posted periodically
  - When providing comments, please note the version
- Elements of Chapter 5 (scale, resolution, time-frequency plane) will be covered after Chapter 6
- Elements of Chapter 4 (certainly sampling) will also be covered later

Finite-Dimensional Uncertainty Principles

Let \( \{x_n\}_{n=0}^{N-1} \) have DFT \( \{X_k\}_{k=0}^{N-1} \).
Let \( N_t \) be the number of nonzero \( x_n \)s.
Let \( N_\omega \) be the number of nonzero \( X_k \)s.

Theorem [Donoho & Stark (1989)]:
For any nonzero signal, \( N_t N_\omega \geq N \).

Corollary: \( N_t + N_\omega \geq 2\sqrt{N} \)

- Sparsity simultaneously in two different bases may be limited
- Is there something special about (standard, DFT) basis pair?
Theorem [Elad & Bruckstein (2002)]:
Suppose nonzero \( x \in \mathbb{R}^N \) has representations
\( x = \Phi \alpha \) and \( x = \Psi \beta \) with
orthonormal bases (matrices) \( \Phi \) and \( \Psi \).
Then \( \| \alpha \|_0 \| \beta \|_0 \geq (\mu(\Phi, \Psi))^{-2} \), where
\( \mu(\Phi, \Psi) = \max_{i,j} | \langle \varphi_i, \psi_j \rangle | \)
\( \mu(\text{std, DFT}) = 1/\sqrt{N} \), so this generalizes previous result
\( \mu(\Phi, \Psi) \geq 1/\sqrt{N} \), so (std,DFT) is a “best” pair

- Sparsity simultaneously in two different bases may be limited
- Is there something special about (standard, DFT) basis pair? Yes

Ramification [Donoho & Stark (1989)]
Suppose true signal \( \{s_n\}_{n=0}^{N-1} \) has DFT-domain
support of \( N_\omega \). Observations of \( s_n \) are made in the
time domain, with only \( M < N \) observations available.
Unique exact recovery is always possible when
\( 2(N - M)N_\omega < N \).
(Also a stability result as seen last time.)

- Sparsity in one basis counteracts missing data when observing
  in the other basis
- Sparser signal implies fewer observations are needed
Simple $\ell^1$-based recovery [Donoho & Stark (1989)]

Suppose true signal $\{s_n\}_{n=0}^{N-1}$ has DFT-domain support of $N_\omega$. Observations of $s_n$ are made in the time domain, with only $M < N$ observations available: $\bar{s} = Ps$, where $P$ sets unobserved components to zero. Under the same condition $2(N - M)N_\omega < N$,

$$\bar{s} = \text{argmin}_{s'} \|s'\|_1 \quad \text{subject to} \quad Ps' = \bar{s}$$

recovers $s$ exactly.

- Sometimes, sparsity can be exploited in a computationally-tractable manner

---

**Recovery of finite sparse signals**

$$y = A$$

Assume $k < m < n$.

Almost any $A$ makes sparsity recoverable from $y$ ($y$ lies in exactly one of the $k$-dimensional subspaces)

Robust to noise? Computationally tractable?
The practitioners’ trick

\[
\begin{array}{ccc}
    y & = & A \\
    m & = & k \\
    n & = & \text{nonzeros}
\end{array}
\]

\[\hat{x} = \arg\min_{\{x : Ax = y\}} \|x\|_2\] is totally wrong (dense)
\[\hat{x} = \arg\min_{\{x : Ax = y\}} \|x\|_0\] is computationally infeasible
\[\hat{x} = \arg\min_{\{x : Ax = y\}} \|x\|_1\] works inexplicably well

A glance at convex relaxation

- We have an underdetermined linear system:
  \[Ax = y\]
  \[x = A^+y\] minimizes \[\|x\|_2\]
  minimizing \[\|x\|_1\], more likely to give a sparse result

The 1-norm is in a precise sense the convexification of the “number of nonzeros” pseudonorm
A road to compressed sensing

- **Superresolution via sparsity** (Donoho, 1992)
  - A signal known to have sparse support can be “stably” recovered from noisy low-frequency (sub-Nyquist) Fourier measurements
  - “Stably” means there is a bound on the noise amplification
  - Nonconstructive

- **Stability of sparsity** (Elad & Bruckstein; Gribonval & Nielsen; Donoho, Elad, & Temlyakov)
  - Optimal sparse approximation of noisy observation has same positions of nonzeros as original vector
  - Coherence: $\mu(A) = \max_{i\neq j} |\langle a_i, a_j \rangle|$
  - Conditions of the form $\|d\| < f(\|x\|, \mu(A))$

- **Heuristics sometimes guaranteed to be exactly right** (Donoho, Elad, & Temlyakov; Gribonval & Nielsen; Tropp)
  - Under conditions on $k$ and $\mu(A)$, basis pursuit and matching pursuit are exact, e.g., $k \leq (3\mu(A))^{-1}$

Compressed sensing

[Candes & Tao (2006)], [Donoho (2006)]

Simplified: Consider $x \in \mathbb{R}^n$ with best $k$-term restriction $\hat{x}_k^{(opt)}$. When

$$\hat{x}_m = \arg\min_{\{x : Ax=y\}} \|x\|_1$$

for suitable $A \in \mathbb{R}^{m \times n}$,

$$\|x - \hat{x}_m\|_2 \leq C \|x - \hat{x}_m^{(opt)}\|_2.$$ 

Random constructions (e.g., i.i.d. Gaussian) for $A$ suffice.

$k \log n$ random measurements and $\ell_1$ recovery are about as good as the $k$ best coefficients!
Sharp result for sparse signals

[Donoho & Tanner (2009)]

Consider $x \in \mathbb{R}^n$ with $k$ nonzero entries.

$$\hat{x}_m = \arg \min_{\{x: Ax = y\}} \|x\|_1$$

recovers $x$ exactly, a.a.s., for i.i.d. Gaussian $A \in \mathbb{R}^{m \times n}$ when $m \approx 2k \log(n/k)$.

Precise knowledge of the cost of using a practical algorithm (factor of $2\log(n/k)$)

What happens with noisy measurements?

Cannot get $x$ exactly, but can recover positions of nonzeros

Orthogonal two-channel filter banks

- “Analysis” maps $x$ to $(\alpha, \beta)$
- “Synthesis” maps $(\alpha, \beta)$ to $x$
- We want this to be an orthonormal expansion
  - Why are analysis and synthesis related by time reversal?
  - What are conditions on $g$ and $h$?
Understanding the synthesis operator

\[ \hat{x}_n = \sum_{k \in \mathbb{Z}} \alpha_k + \sum_{k \in \mathbb{Z}} \beta_k \]

- What are the basis vectors?

- Write the operator with an infinite matrix

- If it is an orthogonal operator, what is its inverse?

- How can \( \{\alpha_k\} \) and \( \{\beta_k\} \) be computed from \( \{x_n\} \)?

Synthesis operator: Matrix form

\[
\begin{bmatrix}
\vdots \\
\hat{x}_0 \\
\hat{x}_1 \\
\hat{x}_2 \\
\hat{x}_3 \\
\hat{x}_4 \\
\vdots \\
\end{bmatrix}
=
\begin{bmatrix}
\ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \\
\vdots & \ldots & g_0 & h_0 & 0 & 0 & 0 & 0 & \ldots \\
\vdots & \ldots & g_1 & h_1 & 0 & 0 & 0 & 0 & \ldots \\
\vdots & \ldots & g_2 & h_2 & g_0 & h_0 & 0 & 0 & \ldots \\
\vdots & \ldots & g_3 & h_3 & g_1 & h_1 & 0 & 0 & \ldots \\
\vdots & \ldots & g_4 & h_4 & g_2 & h_2 & g_0 & h_0 & \ldots \\
\vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \\
\end{bmatrix}
\begin{bmatrix}
\alpha_0 \\
\beta_0 \\
\alpha_1 \\
\beta_1 \\
\alpha_2 \\
\vdots \\
\end{bmatrix}
\]

- If \( \Phi \) orthogonal, what is its inverse?
Analysis and synthesis computations

\[ \alpha_k = \langle g_{n-2k}, x_n \rangle_n \]
\[ \beta_k = \langle h_{n-2k}, x_n \rangle_n \]

\[ \hat{x}_n = \sum_{k \in \mathbb{Z}} \alpha_k g_{n-2k} + \sum_{k \in \mathbb{Z}} \beta_k h_{n-2k}, \]

Making the synthesis orthonormal

• Basis vectors \( \{ g_{n-2k}, h_{n-2k} \}_{k \in \mathbb{Z}} \) must be orthonormal

\[ \langle g_{n-2k}, g_n \rangle_n = \delta_k, \]
\[ \langle h_{n-2k}, h_n \rangle_n = \delta_k, \]
\[ \langle h_{n-2k}, g_n \rangle_n = 0. \]
The lowpass channel

\[ x \rightarrow h_{-n} \rightarrow \alpha \rightarrow g_n \rightarrow x_V \]

**Figure 6.4.** The lowpass branch of a two-channel filter bank, mapping \( x \) to \( x_V \).

- Orthonormality condition can be expressed as
  \[ G(z)G(z^{-1}) + G(-z)G(-z^{-1}) = 2. \]
- Orthonormality condition can also be expressed as
  \[ D_2G^TGU_2 = I \]
- Overall operation \( x \rightarrow x_V \) is an orthogonal projection
  \[ P_V = GU_2D_2G^T \]

The highpass channel

\[ x \rightarrow h_{-n} \rightarrow \beta \rightarrow h_n \rightarrow x_W \]

**Figure 6.5.** The highpass branch of a two-channel filter bank, mapping \( x \) to \( x_W \).

- Orthonormality condition can be expressed as
  \[ H(z)H(z^{-1}) + H(-z)H(-z^{-1}) = 2 \]
- Orthonormality condition can also be expressed as
  \[ D_2H^THU_2 = I \]
- Overall operation \( x \rightarrow x_W \) is an orthogonal projection
  \[ P_W = HU_2D_2H^T \]
Orthogonality of lowpass and highpass channels

- Orthogonality in time domain:
  \[ \langle h_{n-2k}, g_{n-2\ell} \rangle_n = 0 \quad \text{for all } k, \ell \in \mathbb{Z}. \]

- ... in z domain:
  \[ G(z)H(z^{-1}) + G(-z)H(-z^{-1}) = 0. \]

- A way to achieve it (with L even):
  \[ h_n = (-1)^n g_{L-1-n} \iff H(z) = -z^{-L+1}G(-z^{-1}) \]

- Proof:
  \[
  G(z)H(z^{-1}) + G(-z)H(-z^{-1}) \overset{(a)}{=} G(z)(-z^{L-1}G(-z)) + G(-z)((-1)^{-L}z^{L-1}G(z))
  = (-1 + (-1)^{-L}) z^{L-1}G(z)G(-z) \overset{(b)}{=} 0.
  \]

Perfect reconstruction

- Orthogonality of g to even shifts:
  \[ G(z)G(z^{-1}) + G(-z)G(-z^{-1}) = 2. \]

and this choice of h (with L even):

\[ h_n = (-1)^n g_{L-1-n} \iff H(z) = -z^{-L+1}G(-z^{-1}) \]

makes the filter bank an identity operator for all x
Polynomial approximation

\[ x_n \rightarrow [f_n] \rightarrow y_n \]

Def.: If \( y_n = 0 \), then \( f_n \) kills or annihilates \( x_n \)

\( 1 - z^{-1} \) kills any constant

\( 1 - z^{-1} \) reduces degree of a polynomial by one

\( (1 - z^{-1})^{m+1} \) kills polynomials of degree \( \leq m \)

What happens if \( h_n \) kills polynomials of degree \( \leq m \)?

Polynomial approximation in PR FB

Suppose \( h \) kills polynomials

![Diagram showing polynomial approximation in PR FB](image)

Polynomial part conveyed here

"non-smooth details"

Polynomial part conveyed here

"smooth part"
Signal $x$ with projections $x_V$ and $x_W$

![Graphs showing signal projections](image)

**Figure 6.3.** A signal and its two projections. (a) The signal $x$ with various components (low frequency sinusoid, noise, piecewise polynomial, Dirac). (b) The lowpass component $x_V$. (c) The highpass component $x_W$.

**Interlude**

One must learn by doing the thing; for though you think you know it, you have no certainty until you try.

– Sophocles (ca. 496-406 B.C.)

You think you know when you learn, are more sure when you can write, even more when you can teach, but certain when you can program.

– Alan J. Perlis (1922-1990), MIT PhD 1950, winner of the first Turing Award in 1966