Dictionary Learning Using Wavelets

Daniel Mckenzie

University of Georgia

25 April 2016
Overview

This talk is about using wavelets for compression.

The plan:

- Review wavelets.
- Discuss Dictionary learning.
- Explain how one can combine dictionary learning and wavelets to achieve better compression.

I will only consider discrete, finite signals.
Consider the following signals:

- **Figure: Signal A**
- **Figure: Signal B**
- **Figure: Signal C**

Each is a discrete signal consisting of 1024 double precision floating point values, i.e. 65536 bits per signal.
Now consider their (discrete) Fourier Transforms:

![Figure: Signal A](image1)
![Figure: Signal B](image2)
![Figure: Signal C](image3)

Each signal has three non-zero entries, i.e. now 192 bits per signal.

**The Point**  Using Prior knowledge of our signals, we switched basis for $\mathbb{R}^{1024}$ from $\{e_n\}_{n=1}^{1024}$ to $\{\sin(2\pi n)\}_{n=1}^{1024}$. Signals are sparse in new basis and thus can easily be compressed.
Some remarks on Discrete Fourier Transform:

- Linear transformation $\mathbb{R}^N \rightarrow \mathbb{R}^N$, hence can be represented as a matrix: $\hat{f} = Ff$.
- Matrix multiplication takes $O(N^2)$ operations.
- The Fast Fourier Transform takes $O(N \log(N))$.
- ‘The Fast Fourier Transform is the most important numerical algorithm of our lifetime.’ Gilbert Strang. [Str94]
Problem with Fourier Transform

Unable to detect time localized frequency events.

Example (Linear Chirp)

Figure: signal with 4096 entries
Figure: Fourier Transform of signal

Figure: Graph of a Linear Chirp: $y = \sin(2\pi t^2)$ and its Fourier Transform
One Solution:

Instead of basis of sine waves, use a basis of function with compact support in time.

Example (Haar Wavelets ([JJWW00]))

Consider discrete signals of length 8, e.g. \( f = (4, 6, 10, 12, 8, 6, 5, 5) \), in basis \( \{e_i\}_{i=1}^8 \). Let

\[
\varphi = \frac{1}{\sqrt{2}} (1, 1, 0, \ldots, 0)
\]

\[
\psi = \frac{1}{\sqrt{2}} (1, -1, 0, \ldots, 0)
\]

and define translations:

\[
\varphi_k[i] = \varphi[i - 2k]
\]

\[
\psi_k[i] = \psi[i - 2k]
\]
Example (Haar Wavelets cont.)

Then \((\varphi_0, \ldots, \varphi_3|\psi_0, \ldots, \psi_3)\) is a basis. In this basis:

\[ f = \sqrt{2}(5, 11, 7, 5| -1, -1, 1, 0) \]

Figure: \(f\) plotted in red and \(\sqrt{2}(5, 11, 7, 5|0, 0, 0, 0)\) plotted in blue

So \(a_1 = (5, 11, 7, 5)\) is a (very) good approximation to \(f\). Can think of \(a\) as low frequency approximation, \(d_1 = (-1, -1, 1, 0)\) as high frequency details.
Consider translations and dilations:

\[
\varphi_{j,k} = \frac{1}{\sqrt{2^{j+1}}} \varphi \left\lfloor \frac{n - k + 1}{2^j} \right\rfloor
\]

\[
\psi_{j,k} = \frac{1}{\sqrt{2^{j+1}}} \psi \left\lfloor \frac{n - k + 1}{2^j} \right\rfloor
\]

\[
\varphi_{1,0} = (1/2, 1/2, 1/2, 1/2, 0, 0, 0, 0)
\]

Claim \((\varphi_{2,1}, \varphi_{2,2} | \psi_{2,1} \psi_{2,2} | \psi_{1,1}, \ldots, \psi_{1,4})\) is a basis. In this basis:

\[
f = (16, 12 | -6, 2 | -\sqrt{2}, -\sqrt{2}, \sqrt{2}, 0)
\]

\[
a_2 = (16, 12) is still a good approximation to f. \psi is the Haar Wavelet and \varphi is its scaling function.
\]

\[
f \rightarrow (a_1|d_1) is a 1-level Haar Wavelet transform.
\]

\[
f \rightarrow (a_2|d_2|d_1) is a 2-level Haar Wavelet transform.
\]
We shall restrict attention to orthogonal, compactly supported wavelets which come from an MRA, specifically Daubechies wavelets. Some properties:

- Changing to wavelet basis is linear transformation $\mathbb{R}^N \rightarrow \mathbb{R}^N$, hence can be represented as a matrix: $\hat{f} = W_\psi f$.
- Has a ‘Fast Transform’ (Conjugate Mirror Filters).
- Preserves $\ell^2$ norm.
- Is an invertible transformation.
A General Compression Scheme

1. Given 1-dim signal $\mathbf{f}$, take its wavelet transform:

   $\hat{\mathbf{f}} = W_\psi \mathbf{f} = (a_n|d_n| \ldots |d_1)$

2. Either set $d_\ell = 0$ for $\ell = 1, \ldots, k$ or threshold to get $\hat{\mathbf{f}}$, which should be sparse.

3. Encode and store $\hat{\mathbf{f}}$.

4. Reconstruct approximation to $\mathbf{f}$ as needed via: $\tilde{\mathbf{f}} = W_\psi^T \hat{\mathbf{f}}$

**The Point**  
Wavelets are good for (lossy) compression.
Given image $I$, represented as matrix of grayscale values, take an $n$-level wavelet transform to get:

$$\hat{I} = (a_n | h_n, v_n, d_n | \ldots | h_1, v_1, d_1)$$

Set $v_\ell = h_\ell = d_\ell = 0$ for $\ell = 1, \ldots, k$ or threshold to get $\hat{I}$.

Encode and store $\hat{I}$.
512 × 512 fingerprint image analysed with db4 wavelet using MATLAB’s Wavelet Toolbox.
Any Questions?
Recall that an (orthogonal) wavelet transform is like switching to a new basis.

Wavelet transforms suffer from the ‘Curse of Generality’.

**Idea:** Given a collection of ‘similar’ signals \( \{y_1, \ldots, y_N\} \subset \mathbb{R}^n \) (e.g. all fingerprint images of same size). Determine a basis \( \{d_1, \ldots, d_n\} \) of \( \mathbb{R}^n \) s.t. representations of the \( y_i \) in this basis are as sparse as possible.
Definition (Dictionary)

A collection \( \{\mathbf{d}_1, \ldots, \mathbf{d}_K\} \subset \mathbb{R}^n \) which spans \( \mathbb{R}^n \) is a Dictionary. \((K \geq n)\)

Frequently shall write dictionary as \( n \times K \) matrix \( \mathbf{D} \) whose columns are \( \mathbf{d}_i \). The \( \mathbf{d}_i \) are called ‘atoms’.

Definition (Dictionary Learning Problem)

Given \( \{\mathbf{y}_1, \ldots, \mathbf{y}_N\} \) with \( N \gg K \) find a dictionary \( \mathbf{D}^* \) s.t. \( \mathbf{x}_i^* \approx \mathbf{D}^* \mathbf{y}_i \) and the \( \mathbf{x}_i \) are sufficiently sparse. Mathematically:

\[
(\mathbf{D}^*, \mathbf{X}^*) = \arg\min_{\mathbf{D}, \mathbf{X}} \sum_{i=1}^{N} \| \mathbf{y}_i - \mathbf{D} \mathbf{x}_i \|_2^2 \text{ subject to } \| \mathbf{x} \|_0 \leq T
\]

\[
= \arg\min_{\mathbf{D}, \mathbf{X}} \| \mathbf{Y} - \mathbf{D} \mathbf{X} \|_F^2 \text{ subject to } \| \mathbf{x} \|_0 \leq T
\]

If \( \mathbf{Y} \) \( n \times N \) matrix with column \( \mathbf{y}_i \) and \( \mathbf{X} \) a \( K \times N \) matrix with columns \( \mathbf{x}_i \).
Suppose \( T = 1 \), and we require that \( \mathbf{x}_i \) are binary vectors.

Problem becomes:

\[
(D^*, X^*) = \arg\min_{D, X} \|Y - DX\|_F^2 \text{ subject to } \mathbf{x}_i = \mathbf{e}_j \in \mathbb{R}^K \quad (1)
\]
Efficient Algorithm for solution to (1):

Algorithm 1  k-means

Input  \( Y \)
Initialize \( D^{(0)} \in \mathbb{R}^{n \times K} \)

for  \( J = 1 : J_{\text{max}} \) do
  for  \( k = 1 : K \) do
    \( C_k^{(J)} = \{\} \) (an empty list)
  
  for  \( i = 1 : N \) do
    if  \( \| y_i - d_{k^*}^{(J-1)} \|_2 = \min_k \| y_i - d_k^{(J-1)} \| \) then
      Add  \( i \) to list \( C_k^{(J)} \)
    \( C_k^{(J)} \) \hspace{1cm} \( \triangleright \) The Sparse Coding Stage
  
  for  \( k = 1 : K \) do
    \( d_k^{(J)} = \frac{1}{|C_k^{(J)}|} \sum_{i \in C_k^{(J)}} y_i \)
\hspace{1cm} \( \triangleright \) The Dictionary Update Stage

Output  \( D^* = D^{(J_{\text{max}})} \) and \( x_i = e_k \) if  \( i \in C_k^{J_{\text{max}}} \)
The general approach: The K-SVD algorithm [AEB06]

Consider again the general problem:

\[(D, X) = \arg\min_{D, X} \|Y - DX\|_F^2 \text{ subject to } \|x\|_0 \leq T \quad (2)\]

Generalize the previous algorithm to the K-SVD algorithm as follows:

- **Input:** sample signals \(Y = [y_1, \ldots, y_N]\), sparsity parameter \(T\), size of dictionary \(K\).
- Initialize dictionary \(D^{(0)} \in \mathbb{R}^{n \times K}\) with normalized columns.
- Until stopping criteria met, repeat:
  1. **Sparse coding step** Find \(x_i\) such that

     \[x_i = \arg\min_{x} \{\|y_i - Dx\|_2\} \text{ subject to: } \|x\|_0 \leq T \quad (3)\]

     Using MP, OMP etc.
  2. Let \(X \in \mathbb{R}^{K \times N}\) have columns \(x_i\).
(Dictionary Update) Update each atom $d_k$ in turn:

- Let $\omega_k$ be the set of examples that use $d_k$: $\omega_k = \{j : (x_j)_k \neq 0\}$.
- For each $j \in \omega_k$ compute $e_{j,k} = y_j - \sum_{\ell, \ell \neq k} (x_j)_\ell d_\ell$, residual without $d_k$.
- We are going to update $d_k$ and the coefficients $(x_\ell)_k$ (for $\ell \in \omega_k$) so as to minimize this residual error $E_k$.
- Let $E_k \in \mathbb{R}^{n,|\omega_k|}$ have columns $e_{j,k}$.
- Solve:

$$(d_k^*, \xi^*) = \arg\min_{\xi, \mathbf{d}} \|E_k - \mathbf{d} \xi\|_F^2 \text{ subject to: } \xi \in \mathbb{R}^{|\omega_k|} \text{ and } \|\mathbf{d}\|_2 = 1$$

(4)

- $d\xi \in \mathbb{R}^{n,|\omega_k|}$ is rank one. so (4) is solved by choosing $d\xi$ to be optimal rank one approximation to $E_k$.
- (by Eckart-Young-Mirsky theorem) if $U\Sigma V^T = E_k$ is SVD, then $d_k^* = u_1$ and $\xi^* = \sigma(1)v_1$ solves (4).
- Update $d_k$ to $d_k^*$ and $(x_{j_\ell})_k = \xi_\ell$ where $\omega_k = \{j_1, j_2, \ldots\}$.
Output: Learned dictionary $\mathbf{D}$ and sparse representation matrix $\mathbf{X}$ such that if $\mathbf{X}$ has columns $\mathbf{x}_i$ then $||\mathbf{x}_i||_0 \leq T$ and $\mathbf{D}\mathbf{x}_i \approx \mathbf{y}_i$
Some Remarks

- Updating only coefficients \((x_j)_k\) for \(j \in \omega_k\) ensures that sparsity of the \(x_i\) can only improve.
- If in sparse coding step (3) solution is always found, then representation error \(|Y - DX|^2_F\) is non-increasing, thus algorithm will converge.
- OMP works well enough for fairly small \(T\) to ‘practically’ ensure convergence.
In [AEB06] K-SVD is implemented for images of faces. 

$N = 11\,000$ and the $y_i$ for $i = 1, \ldots, K$ are $8 \times 8$ pixel blocks (i.e. $n = 64$), randomly sampled from a database of $4752 \times 4752$ facial images.

K-SVD run with $K = 441$

**Figure**: Learned dictionary on right, Haar dictionary on left (from [AEB06])

For further details see [AEB06]
Experimental Results

- I randomly sampled image from same database, split into 594 $8 \times 8$ blocks $B_i$.
- Fix # bits per coefficient, $Q$.
- Fix an error goal $\epsilon$.
- Encode each $B_i$ to $\tilde{B}_i$ using OMP such that if $e^2 = \frac{||B_i - \tilde{B}_i||^2}{64}$ then $e^2 < \epsilon$.
- Let $PSNR = 10 \log_{10}(\frac{1}{e^2})$ (higher PSNR = better quality).
- Let $TNB$ denote the total number of bits required to encode $\tilde{I}$.

$$TNB = \#\text{blocks} \times a + \#\text{coeffs}(b + Q)$$

where $a$ is # bits required to code # coefficients per block, $b$ is # bits required to code index of each atom.

- BPP (Bits Per Pixel) is given by:

$$BPP = \frac{TNB}{\#\text{pixels}}$$
**Figure:** BPP vs. PSNR for learned dictionary, Haar wavelet transform and DCT. From [AEB06]
Figure: Reconstruction of compressed images. From [AEB06]
Learned dictionaries appear to offer better compression rates, at least at low PSNR. However:

1. Learning is computationally intensive.
2. Lack of fast transform
3. Not multiscale; loses out on some potential compression.

**Question:** Can we combine strengths of learned and wavelet dictionaries?
Any questions?
[OLE11] attempts to combine a wavelet transform with a learned dictionary.

**Idea:** First take wavelet transform, then apply a dictionary learning algorithm (K-SVD).

Formally, solve:

$$(\tilde{D}^*, X^*) = \arg\min_{\tilde{D}, X} \|Y - W_\psi \tilde{D}X\|_F^2 = \arg\min_{\tilde{D}, X} \|W_\psi Y - \tilde{D}X\|_F^2$$

where the columns of $\tilde{D}$ are constrained to be very sparse.

**Effective dictionary is now** $D = W_\psi \tilde{D}$ whose atoms are linear combinations of several wavelet atoms, adapted to training set $Y$. 

Implementation

- In [AEB06] a 3 layer db4 wavelet transform was taken on data base of 20 coastal scenery images.
- 3 levels gives 10 bands: \( a_3, d_3, v_3, h_3, \ldots, h_1 \). Will train a dictionary \( D_b \) for each band (10 in total).
- For each band, take \( K = 64 \) (# of atoms), and again each atom will be a \( 8 \times 8 \) block.
- As before, train \( D_b \) using a training set \( Y = [y_1, \ldots, y_N] \) of \( 8 \times 8 \) blocks randomly drawn from b-th band of images using K-SVD.
Given image \( I \), take 3 level wavelet transform \( \hat{I} \).

For each band \( b \), split \( \hat{I}_b \) into \( 8 \times 8 \) blocks \( B_i \).

encode each block \( B_i \) to \( \tilde{B}_i \) using OMP such that in total only \( M \) atoms are used.

This is compared to Wavelet transform compression using thresholding to keep only the \( M \) largest coefficients, and to regular K-SVD using \( M \) coefficients.
Figure: Comparing PSNR to $M$ for three methods. [OLE11]
**Figure:** reconstructions using 32 000 terms. Top left is original, top right is wavelet (db4) reconstruction, bottom left is regular K-SVD, bottom right is K-SVD + Wavelet. [OLE11]
Conclusions

To conclude:

- In order to compress one first needs to choose a dictionary such that signal becomes sparse.
- ‘generic dictionaries’ (Wavelets etc.) provide fast transforms, but are not adapted to signals at hand.
- ‘learned dictionaries’ provide good sparsity, but lack of fast transform can be prohibitive.
- learning a dictionary whose atoms are made up of generic atoms may allow one to squeeze out some extra sparsity by adapting to a given class of signals.
Thank You!

