Sparse Random Wavelet Representation & Decomposition

Nicholas Richardson[1], Giang Tran[1], Hayden Schaeffer[2], and Rachel Ward[3]

[1]University of Waterloo, [2]Carnegie Mellon University, [3]The University of Texas at Austin

Summary

- Novel method for representing and decomposing 1D signals
- Represent signal in a random wavelet "basis"
- Find representation via Spectral Projected Gradient with L1 minimization (SPGL1)
- Decompose representation into modes via Density-Based Spatial Clustering of Applications with Noise (DBSCAN) or frequency filters

Problem Statement

- Given a 1D signal y, find a sparse representation
- Decompose it into simple modes

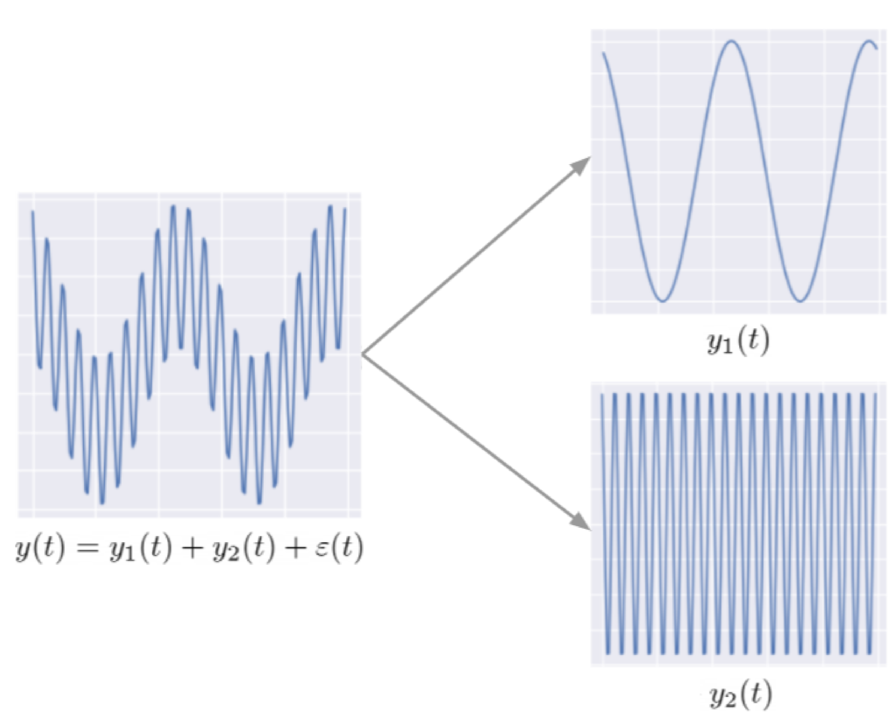

Figure 1: Decomposition Visualisation

Wavelet "Basis"

$$g(t) = e^{-2\frac{(t-\tau)^2}{w^2}} \sin(2\pi f \cdot t + \phi),$$
$$(\tau, f, \phi) \in \mathcal{U}_{[0,t_{\max}]\times[0,f_{\max}]\times[0,2\pi]}$$

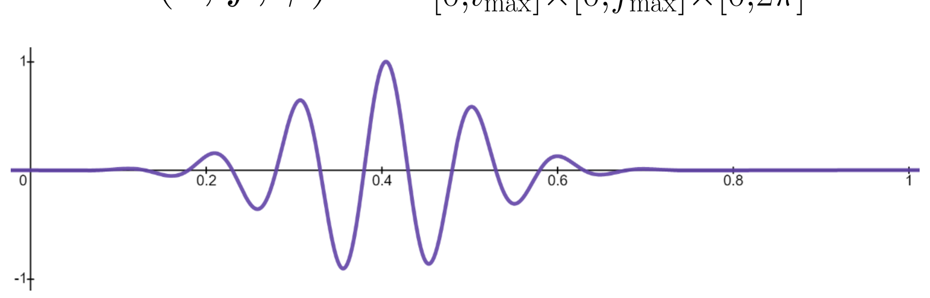

Figure 2: Example real-valued Gabor wavelet with $(\tau, f, \phi; w) = (0.4, 10, 1.26; 0.2)$.

Representation Algorithm

input: $t, y \in \mathbb{R}^N, M \in \mathbb{Z}^+, w, f_{\max} \in \mathbb{R}^+,$
$r \in]0,1[$
Generate $\{\tau_m, f_m, \phi_m\}_{m=1}^M$;
Generate M wavelets $g_m \in \mathbb{R}^N$;
Store wavelets by column in a matrix $G \in \mathbb{R}^{N\times M}$;
Solve
$x^* = \arg\min_{x\in\mathbb{R}^M} \|x\|_1 \text{ s.t. } \|Gx - y\|_2 < r\|y\|_2$;
output: $x^*, G, \{\tau_m, f_m, \phi_m\}_{m=1}^M$

- Reconstruct $y \approx Gx^* = \sum_{m=1}^{M} x_m g_m$
- Relative error $\frac{\|Gx^* - y\|_2}{\|y\|_2}$ is less than r

Mathematical Example

Input: $y(t) = y_1(t) + y_2(t) + y_3(t)$ defined by
$y_1(t) = \pi t,\ t \in [0, 5/4[$
$y_2(t) = \cos(40\pi t),\ t \in [0, 5/4[$
$y_3(t) = \cos\left(\frac{4}{3}\left((2\pi t - 10)^3 - (2\pi - 10)^3 + 20\pi(t-1)\right)\right)$
$t \in]1, 2].$
Parameters: $f_{\max} = 80, M = 16\,000, r = 5\%,$
$w = 0.1\,\text{s}, \texttt{min\_samples} = 3, \epsilon = 0.1, s = 1/80.$

"Synchrosqueezed wavelet transforms: An empirical mode decomposition-like tool"; Daubechies, Lu, & Wu; Applied and Computational Harmonic Analysis, 2011. DOI:10.1016/j.acha.2010.08.002

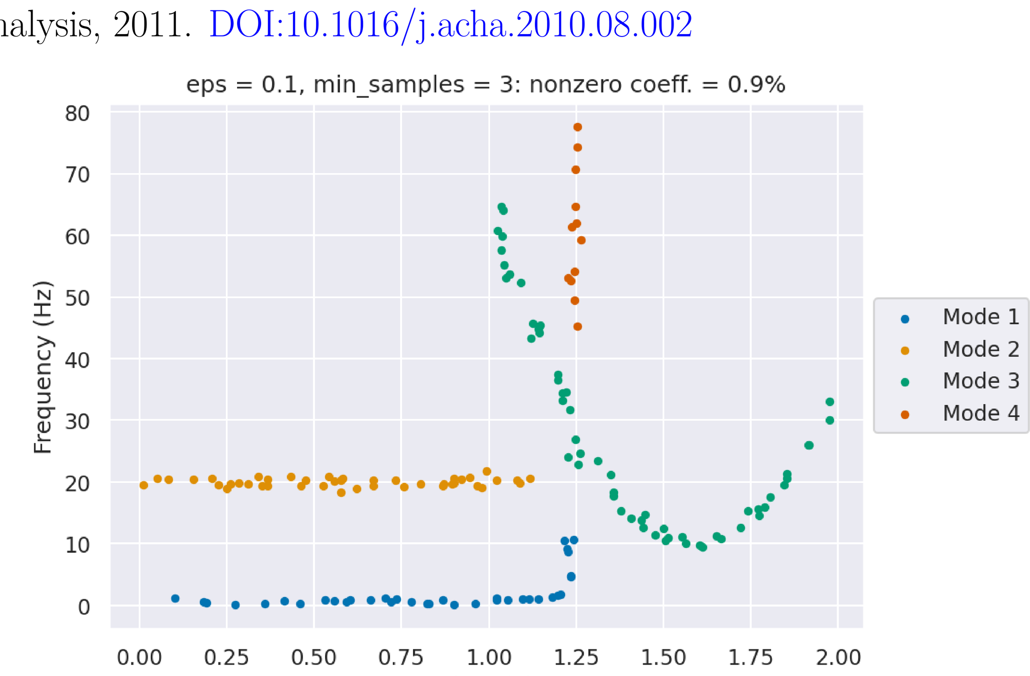

Figure 3: Segmentation of nonzero wavelets into four modes. Frequency and time-shifts of nonzero wavelets are plotted where colour represents labelling from DBSCAN.

Decomposition Algorithm

input: $x^*, G, (\tau_m, f_m)_{m=1}^M,$
$\texttt{min\_samples} \in \mathbb{Z}^+, \epsilon, s \in \mathbb{R}^+$
Define $(\tau_{m_j}, f_{m_j})_{j=1}^{M'} := \{(\tau_m, f_m) | x_m^* \neq 0\}$;
Scale input points to obtain $(\tau_{m_j}, s \cdot f_{m_j})$;
Use DBSCAN to label each point by cluster to obtain $\{\ell_{m_j}\}$ where $\ell_{m_j} \in \{-1, 0, \cdots, K-1\}$;
Extract K modes:
$y_k = \sum_{m\in I_k} x_m^* g_m,\ I_k = \{m_j | \ell_{m_j} = k-1\}$;
output: $\{y_k\}_{k=1}^{K'}$

- Alternatively decompose using known conditions
- Ex. a band-pass filter could be represented as $I = \{m_j | a < f_{m_j} < b\}$

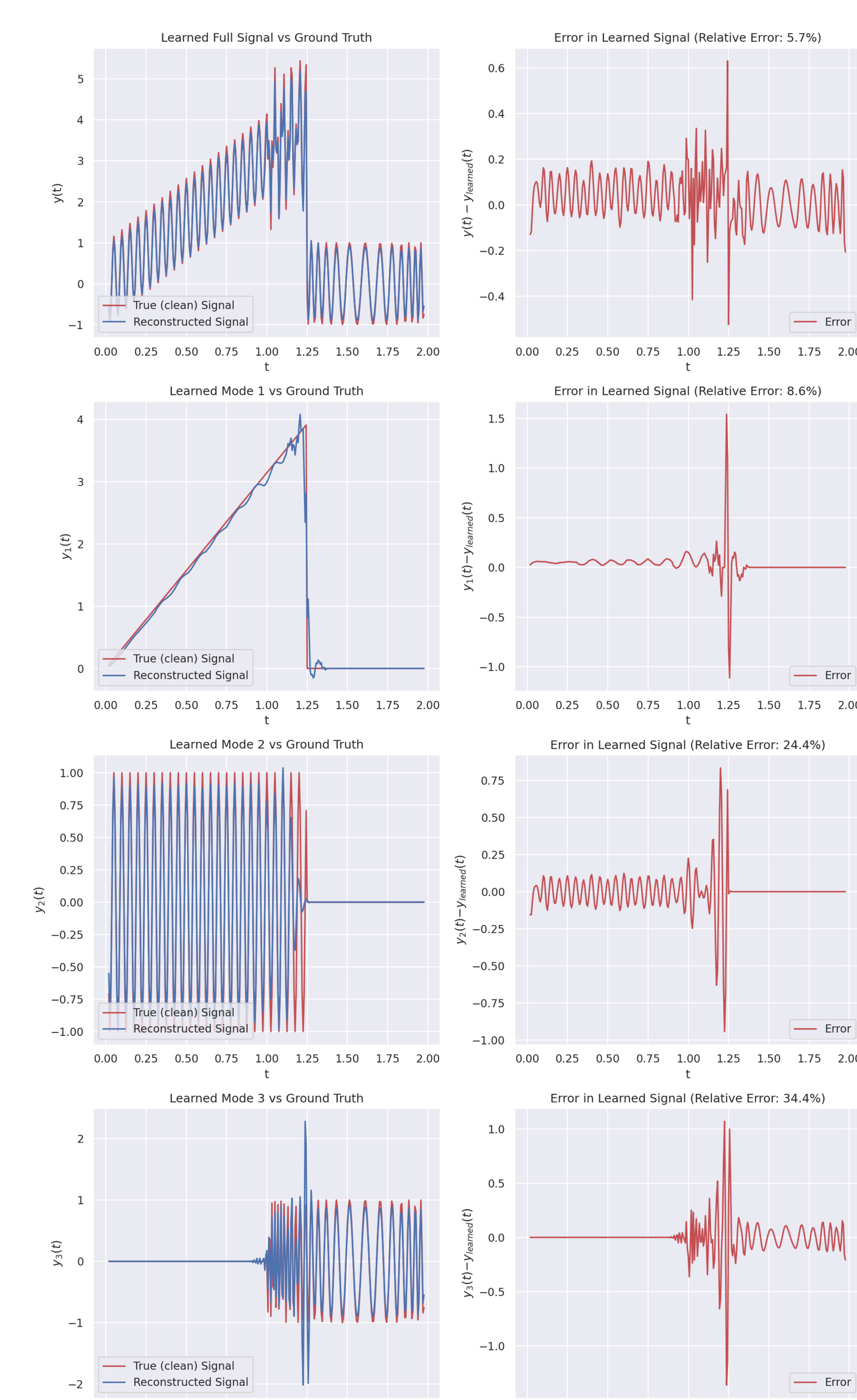

Figure 4: Learned signal and decomposed modes vs ground truth. Note input was a noisy source (5% Gaussian), and mode 4 was added to mode 1 in the second row.

Flute and Guitar Decomposition

Input: $y(t) = y_1(t) + y_2(t)$ where $y_1(t)$ is a flute and $y_2(t)$ is a guitar clip around 1.85 s long.
Parameters: $f_{\max} = 44\,100\,\text{Hz}/16, M = 51\,080,$
$N = 10\,200, r = 8\%, w = 0.03\,\text{s}.$

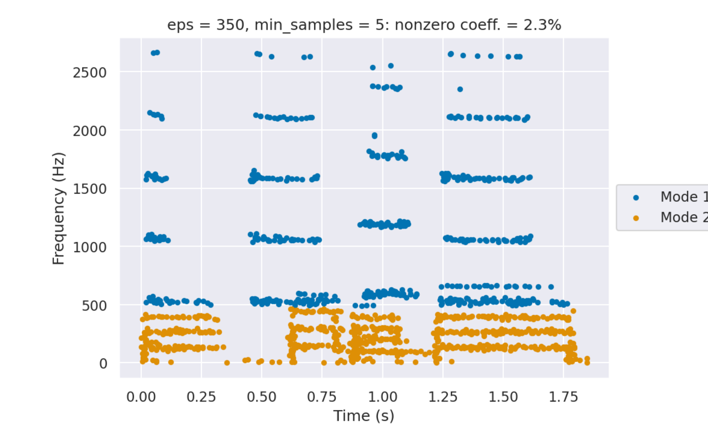

Figure 5: Segmentation via slice at $f = 480\,\text{Hz}$

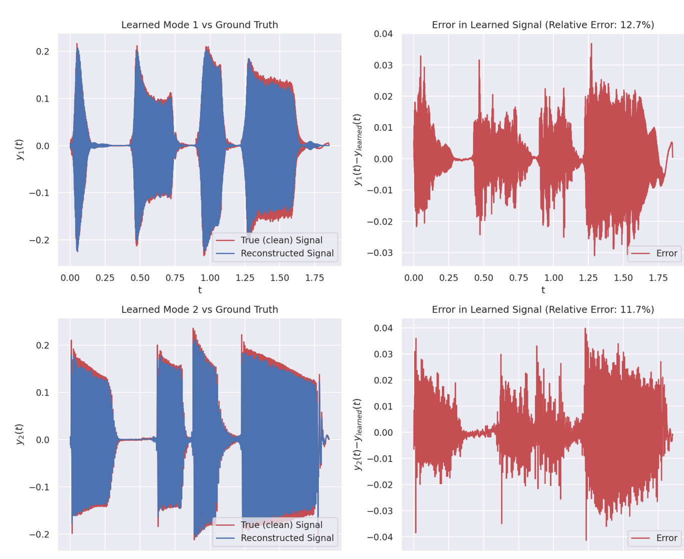

Figure 6: Extracted flute and guitar vs original

References

1. Purwins, H. *et al.* Deep Learning for Audio Signal Processing. *CoRR* **abs/1905.00078.** arXiv: 1905.00078 (2019).
2. Hashemi, A. *et al. Generalization Bounds for Sparse Random Feature Expansions.* 2021. arXiv: 2103.03191.
3. Van den Berg, E. & Friedlander, M. P. Sparse Optimization with Least-Squares Constraints. *SIAM Journal on Optimization* **21,** 1201–1229 (2011).
4. Louhichi, S., Gzara, M. & Ben Abdallah, H. *A density based algorithm for discovering clusters with varied density.* in *2014 World Congress on Computer Applications and Information Systems (WCCAIS)* (2014), 1–6.

Contact Information

Nicholas Richardson: njericha@uwaterloo.ca
Code: will be available on Github soon