Compressive Sampling in Perceptual Audio Compression Proposal

Stephen Pinto

May 15, 2015

1 Background

Perceptual Audio Compression (PAC) is a generic term for how dozens of prevalent audio codecs (e.g. MPEG and AAC) achieve impressive compression ratios for audio files. In a sentence, it is any lossy compression algorithm which seeks to compress audio signals at the expense of perfect reconstruction while maintaining good fidelity by human hearing standards. The algorithms seek to place quantization noise inherent in compression into spectral regions that are difficult for humans to hear. Developed in the early 90s, PAC has since been much improved and become ubiquitous. For an in-depth treatment on the basics of a perceptual audio codec, see [BG03].

Compressive Sampling (CS) is a computational technique for data compression and reconstruction which centers around a convex optimization problem. Under certain conditions (thoroughly discussed in a series of papers by Candes and Tao among others and summarized in [CW08]), a signal that is sparse in some domain can be perfectly reconstructed from a compressed set of samples. CS is perhaps most well known for its application in the field of medical imaging.

CS has seen minimal applications in the world of audio applications ([DMD$^+$13] being one great example for the purposes of declipping audio signals) and even fewer still to audio compression specifically (I have found none.) This proposed project will apply CS to a simple perceptual audio codec. Achieving compression ratios as high as cutting edge audio codecs is unreasonable, but achieving modest compression ratios with this novel technique – at least to the world of audio compression – is an achievable goal.

2 Proposal

The final project EE 364b will overlap with 3 credits of Independent Study under Prof. Marina Bosi explicitly meant to implement this CS based perceptual audio codec. The project is already underway, currently utilizing CVXPY, and extremely slow.

The codec takes in a signal and splits it into overlapping blocks of a constant length (typically 512 or 1024 samples). Each block is encoded and subsequently decoded independently. As such, the following description of the encoding and decoding stages is applied to each individual block one by one.

The encoding half of the codec takes in a signal f of length n (the block) and outputs a signal y of length $m \ll n$, calculated as

$$y = \Phi H f.$$

The sensing basis, $\Phi \in \mathbb{R}^{m \times n}$, has rows randomly selected from the standard basis of $\mathbb{R}^n$. The perceptual weighting matrix, $H \in \mathbb{R}^{n \times n}$, is a circulant matrix diagonalized by the Discrete Cosine Transform matrix with eigenvalues equal to the inverse of the signal's masking threshold. This masking threshold indicates the necessary magnitude a signal must contain at a certain frequency for humans to hear it. The interested reader is again referred to [BG03] for details on calculating the masking threshold.

In the simplest case, the decoding stage receives the encoded signal y and forms a reconstruction of the original signal, $\hat{f}$, by solving the convex optimization problem

$$\begin{aligned} \hat{x} = \underset{x}{\text{argmin}} \quad & \|x\|_1 \\ \text{subject to} \quad & \|y - \Phi H \Psi x\|_2 \leq \kappa, \end{aligned} \tag{1}$$

where $\Psi \in \mathbb{R}^{n \times n}$ is the sparsifying matrix (the Inverse Discrete Cosine Transform matrix here), making $\hat{f} = \Psi \hat{x}$, and κ is some constant fidelity factor. However, Candes et al. present an improved reconstruction method in [CWB08] which changes the reconstruction of a single block to an iterative process where each iteration solves the convex optimization problem

$$\begin{aligned} \hat{x}^{(i)} = \underset{x}{\text{argmin}} \quad & ||W^{(i)} x||_1 \\ \text{subject to} \quad & ||y - \Phi H^{(i)} \Psi x||_2 \leq \kappa. \end{aligned} \tag{2}$$

where

$$W^{(i+1)} = \text{diag}\left(\frac{1}{|\hat{x}_1^{(i)}| + \delta}, ..., \frac{1}{|\hat{x}_n^{(i)}| + \delta} \right)$$

for some small constant δ and $H^{(i+1)}$ has eigenvalues equal to the inverse of the masking threshold of x^i. This iteration continues until $\hat{x}$ converges (i.e. $||\hat{x}^{(i)} - \hat{x}^{(i-1)}||_2 \leq \gamma$ for some γ - this typically takes less than 4 iterations.)

Implemented in CVXPY, the net computation time for all iterations over all blocks over all audio channels is unreasonably large (tens of minutes) – especially for realistically sized audio samples. Fortunately, the reconstruction process involves matrices with plenty of structure - W is diagonal, H is circulant, and Ψ is orthogonal. For this EE364b project, I will implement a custom solver in Python that takes advantage of this structure to greatly speed up the codec.

Acknowledgments

Prof. Marina Bosi is formally advising this project under 3 credits of Independent Study (EE 391.)

References

[BG03] Marina Bosi and Richard E. Goldberg. *Introduction to Digital Audio Coding and Standards*. Kluwer Academic Publishers, 2003.

[CW08] Emmanuel J. Candes and Michael B. Wakin. An Introduction To Compressive Sampling. *IEEE Signal Processing Magazine*, 25(March 2008):21–30, 2008.

[CWB08] Emmanuel J. Candes, Michael B. Wakin, and Stephen P. Boyd. Enhancing sparsity by reweighted 1 minimization. *Journal of Fourier Analysis and Applications*, 14:877–905, 2008.

[DMD$^+$13] B Defraene, N Mansour, S De Hertogh, T van Waterschoot, M Diehl, and M Moonen. Declipping of Audio Signals Using Perceptual Compressed Sensing. *Audio, Speech, and Language Processing, IEEE Transactions on*, 21(12):2627–2637, 2013.