Indian Institute of Technology, Madras

Department of Electrical Engineering

A Compressed Sensing Approach for Denoising Audio and Speech Signals

Convex Optimization - EE5121 - End Semester Paper

Name: Vallabh Ramakanth

Roll No.: EE17B068

Date: July 26, 2020

Main Reference: Dalei Wu, Wei-Ping Zhu and M.N.S. Swamy.
A Compressive Sensing Method for Noise Reduction of Speech and Audio Signals.
2011 IEEE 54th International Midwest Symposium on Circuits and Systems (MWSCAS).

Contributions: This paper contributes by:

- Modelling the optimisation problem differently to avoid weighing of L2 and L1 objectives.
- Working with sparsity in Fourier (STFT) domain instead of Daubechies Transform domain.
- Implementing the algorithm on voice signals with higher sampling frequency.

1 Brief on Compressed Sensing

The classical Nyquist sampling theorem states that every signal with a maximum N frequency components can be perfectly recovered with a sampling rate of at least 2N. Compressed Sensing (CS) is a method of almost perfect signal recovery of an undersampled signal provided that the signal is $k-$sparse in some domain. This has great consequences in many fields as a large variety of real-life signals are sparse in some transform domain. CS also allows systems to sample at a rate lower than the Nyquist sampling limit 2N to recover sparse signals having frequency components larger than N (as in [1]).

However, the theoretical framework of CS relies on certain properties of the measurement matrix and the sparsity of the signal to be estimated. They are as follows (as in [1]):

1. **$k-$sparsity of signal space:** The signal space $\mathcal{S}$ holds $k-$sparsity if $\forall x \in \mathcal{S}, ||x||_0 \leq k$.
2. **Restricted Isometry Property (RIP):** The measurement matrix A should satisfy RIP, i.e., for measurement matrix $A_{m\times n}, k << m < n$, $\exists \delta_k > 0$ such that $\forall x$

$$(1-\delta_k)||x||_2^2 \leq ||Ax||_2^2 \leq (1+\delta_k)||x||_2^2$$

Then CS theory tells us that the vector can be almost perfectly recovered from

$$x^* = \text{argmin}_z ||z||_1 \text{ subject to } Az = y$$

where y is the true m length measured vector and

$$||x - x^*||_2 \leq C\frac{\sigma_k(x)_1}{\sqrt{k}}$$

where x is the signal to be estimated, $C > 0$ is some constant and $\sigma_k(x)_1 = \inf_{s\in\mathcal{S}} ||x - s||_1$ is the best $k-$sparse approximation of x.
One good family of measurement matrices which satisfy the above mentioned property are Random Partial Fourier Matrices (RFPMs) and we will only deal with them.

2 Voice as a Sparse Signal

Human voice and speech is very dynamic, comprising frequencies, amplitudes and tones which vary largely with time. The assumption made is that the frequency content of speech is stationary in a frame of time length $\approx$ 100ms. In a given frame, we hypothesise that the frequency content is stationary and there are only k dominant frequencies.
Hence, the human voice is treated as a $k-$sparse vector in the frequency domain.

$$v(t) \xrightarrow{\text{sampling}} \mathbf{v}[n] \xrightarrow{\text{FFT}} \mathbf{V}[j]$$

$\mathbf{V}[j]$ is the n length frequency vector which is approximately $k-$sparse. On the contrary, noise is not sparse in frequency domain. Hence, getting the best sparse representation of a noisy audio clip using CS can facilitate denoising the signal.

3 Modelling of the Problem

Let $\mathbf{s}$ be the noisy n length audio sample of the frame. $\mathbf{y} = M\mathbf{s}$ is the "measured vector" where $M_{m\times n}$ is a matrix which randomly chooses rows (in order) in $\mathbf{s}$.

$$M = \begin{pmatrix} 0 & \dots & 1 & \dots & 0 \\ 0 & \dots & 0 & 1 & \dots \\ \vdots & & \ddots & & \vdots \\ 0 & \dots & 0 & \dots & 1 \end{pmatrix}_{m\times n}$$

Let $\mathbf{x} \in \mathbb{C}^n (\equiv \mathbb{R}^{2n})$ be the $k-$sparse frequency vector.
Then

$$\mathbf{s} = F\mathbf{x} + \epsilon \tag{1}$$

$$\mathbf{y} = MF\mathbf{x} + M\epsilon = A\mathbf{x} + M\epsilon \tag{2}$$

where F is the full inverse Fourier matrix ϵ is the noise term. $A = MF$ is the measurement matrix. Therefore,

$$||\mathbf{y} - A\mathbf{x}||_2^2 \leq ||M\epsilon||_2^2 \leq ||\epsilon||_2^2 \tag{3}$$

If we define γ as the output signal to noise ratio, then

$$\begin{aligned}\gamma &= \frac{||F\mathbf{x}||_2^2}{||\epsilon||_2^2} \\ &= \frac{1}{n}\frac{||\mathbf{x}||_2^2}{||\epsilon||_2^2} \qquad (\because F^H F = \frac{1}{n} I)\end{aligned}$$

γ could also be considered as a "fitting parameter". Constraint (3) becomes

$$\mathbf{y}^T\mathbf{y} - 2\mathbf{y}^T A\mathbf{x} + \mathbf{x}^H(A^H A)\mathbf{x} - \frac{\mathbf{x}^H\mathbf{x}}{n\gamma} \leq 0 \tag{4}$$

This is a quadratic constraint. From CS method, our required vector is the solution of the convex optimisation problem:

$$\begin{aligned}&\min_{\mathbf{x}\in\mathbb{C}^n(\equiv\mathbb{R}^{2n})} \quad ||\mathbf{x}||_1 \\ &\text{subject to} \quad \mathbf{y}^T\mathbf{y} - 2\mathbf{y}^T A\mathbf{x} + \mathbf{x}^H(A^H A)\mathbf{x} - \frac{\mathbf{x}^H\mathbf{x}}{n\gamma} \leq 0\end{aligned}$$

We now need to show that Slater's Rule holds so that we can conclude strong duality of the problem. Consider

$$\mathbf{x}^H(A^H A)\mathbf{x} = \mathbf{x}^H(F^H M^T M F)\mathbf{x} \leq \mathbf{x}^H(F^H F)\mathbf{x} = \frac{\mathbf{x}^H\mathbf{x}}{n} \tag{5}$$

$$\implies \mathbf{x}^H(A^H A)\mathbf{x} - \frac{\mathbf{x}^H\mathbf{x}}{\gamma} \leq \mathbf{x}^H\mathbf{x}\left(\frac{1}{n} - \frac{1}{n\gamma}\right) = c\,\mathbf{x}^H\mathbf{x} \tag{6}$$

$$\implies c\left(\frac{\mathbf{y}^T\mathbf{y}}{c} - 2\left(\frac{A^H\mathbf{y}}{c}\right)^H \mathbf{x} - \mathbf{x}^H\mathbf{x}\right) = c\left(\left|\left|\mathbf{x} - \frac{A^H\mathbf{y}}{c}\right|\right|_2^2 + \frac{1}{c}\,\mathbf{y}^T\left(I_m - \frac{AA^H}{c}\right)\mathbf{y}\right) \tag{7}$$

Now, if we show that $c > 0$ and $I_m - \frac{AA^H}{n} \prec 0$, we would have successfully shown that $\exists\mathbf{x} \in \text{relint}(\mathbb{C}^n) = \mathbb{C}^n$ which satisfies the quadratic constraint (4). So,

$$\begin{aligned}AA^H = MFF^H M^T = \frac{MM^T}{n} = \frac{I_m}{n} \\ \implies I_m - \frac{AA^H}{c} = I_m\left(1 - \frac{1}{nc}\right)\end{aligned}$$

We have shown that we are guaranteed that $\exists\mathbf{x} \in \text{relint}(\mathbb{C}^n) = \mathbb{C}^n$ which satisfies constraint (4) if $c > 0$ and $1 - \frac{1}{nc} < 0$. The problem is guaranteed to satisfy Slater's Rule and is convex $\forall\mathbf{y}$ if we choose $\gamma > 1$.
Hence, the dual and the primal problem have the same optimal value as strong duality is valid. The dual problem is as follows.

$$\begin{aligned}&\max_{\lambda\in\mathbb{R}} \quad \inf_{\mathbf{x}\in\mathbb{C}^n} ||\mathbf{x}||_1 + \lambda\left(\mathbf{y}^T\mathbf{y} - 2\mathbf{y}^T A\mathbf{x} + \mathbf{x}^H(A^H A)\mathbf{x} - \frac{\mathbf{x}^H\mathbf{x}}{n\gamma}\right) \\ &\text{subject to} \quad \lambda \geq 0\end{aligned}$$

This can be solved using simultaneous gradient descent (for $\mathbf{x}$) and gradient ascent (for λ) (as described in [4]). Algorithm 1 is used to achieve the optimal solution.

Note 1: Since we are dealing with functions which map $\mathbb{C}^n \to \mathbb{R}$, it is essentially a map from $\mathbb{R}^{2n} \to \mathbb{R}$, and we can define a gradient of the function with respect to $\mathbf{x}$ using Wirtinger Calculus.

Note 2: $\mathbf{s}$ is a real vector, and hence the DFT vector will be conjugate symmetric and hence we can reduce the search space from $\mathbb{C}^n$ to $\mathbb{C}^{\text{ceil}(n/2)}$ and tightly constrain the system. However for simplicity of implementation and to maintain convexity of the problem, we will not truncate $\mathbf{x}$.

Note 3: For numerical implementation of gradient descent, we can define a function (as in [5])

$$shrink(\mathbf{x}) : \mathbb{C}^n \to \mathbb{C}^n$$
$$shrink(\mathbf{x})_i = \begin{cases} \frac{x_i}{|x_i|} : |x_i| > \eta \\ 0 \quad : |x_i| \leq \eta \end{cases}$$

where $\eta \to 0$. This gives us the gradient of $||\mathbf{x}||_1$ in $\mathbb{C}^n \setminus \{\mathbf{x}|x_i = 0\}$. However, it is one of the sub-gradients[6] of $||\mathbf{x}||_1$ and will work in the gradient descent algorithm to move towards the optimal point. We set η to 10^{-9}.

At the end of the algorithm, a threshold for sparsity is added, i.e., x_{ij} is made 0 if $|x_{ij}| < 0.1$.

Algorithm 1 Simultaneous Gradient Descent and Ascent with Backtracking Line Search ($\gamma = 10^4$ or 40dB)

```
$\delta := 10^{-8}$  (tolerance)
$\alpha := 0.1$  (line search parameter)
$\beta := 0.5$  (line search parameter)
$f_0(\mathbf{x}, \lambda) := ||\mathbf{x}||_1 + \lambda \left( ||\mathbf{y} - A\mathbf{x}||_2^2 - \frac{1}{n\gamma}||\mathbf{x}||_2^2 \right)$
$\lambda_0 \leftarrow 0$
$\mathbf{x}_0 \leftarrow$ FFT of frame
repeat
  $\Delta\mathbf{x}_i \leftarrow \nabla_{\mathbf{x}_i} f_0 = shrink(\mathbf{x}_i) + \lambda_i((2A^H(\mathbf{y} + A\mathbf{x}_i)) - \frac{2}{n\gamma}\mathbf{x}_i)$
  $t_x \leftarrow 1$
  repeat
    $t_x \leftarrow \beta t_x$
  until $f_0(\mathbf{x}_i + t_x \Delta\mathbf{x}_i, \lambda_i) < f_0(\mathbf{x}_i, \lambda_i) - \alpha t_x ||\Delta\mathbf{x}_i||_2^2$
  $\mathbf{x}_{i+1} \leftarrow \mathbf{x}_i - t_x \Delta\mathbf{x}_i$
  $\Delta\lambda_i \leftarrow \nabla_{\lambda_i} f_0 = ||\mathbf{y} - A\mathbf{x}_{i+1}||_2^2 - \frac{1}{n\gamma}||\mathbf{x}_{i+1}||_2^2$
  $t_\lambda \leftarrow 1$
  repeat
    $t_\lambda \leftarrow \beta t_\lambda$
  until $f_0(\mathbf{x}_{i+1}, \lambda_i + t_\lambda \Delta\lambda_i) > f_0(\mathbf{x}_{i+1}, \lambda_i) + \alpha t_\lambda \Delta\lambda_i^2$
  if $\lambda_i + t_\lambda \Delta\lambda_i \geq 0$ then
    $\lambda_{i+1} \leftarrow \lambda_i + t_\lambda \Delta\lambda_i$
  else
    $\lambda_{i+1} \leftarrow \lambda_i$
  end if
until $|x_{i+1,j} - x_{i,j}| < \delta$ or $i = 1000$  (from KKT conditions, if $\mathbf{x}$ is the minimum, then $\lambda$ is the maximum)
```

4 Implementation

The above denoising technique was tested using the Python programming language. The audio clips are single channel 16-bit `.wav` files with sampling frequency 16kHz. The frame size chosen was $n = 1024$ and the sampling size was $m = 256$.

Following are snippets of the code implemented in python. To prevent full matrix multiplication and increase speed of the program, array slicing and the FFT algorithm were used instead of sampling matrices and the complete Fourier matrix.

```python
#   lm ==> lagrange multiplier
#   ii ==> the m indices to be sampled
def f(x, y, ii, lm):
  # Objective function
  xt = np.fft.ifft(x)
  vec = xt[ii]
  T1 = np.sum((np.abs(vec - y))**2)
  T2 = np.sum(np.abs(x)**2)/n/snr
  T3 = np.sum(np.abs(x))
  return lm*(T1 - T2) + T3

def grad_f_x(x, y, ii, lm):
  # Gradient of objective wrt complex x
  xt = np.fft.ifft(x)
  vec = xt[ii]
```

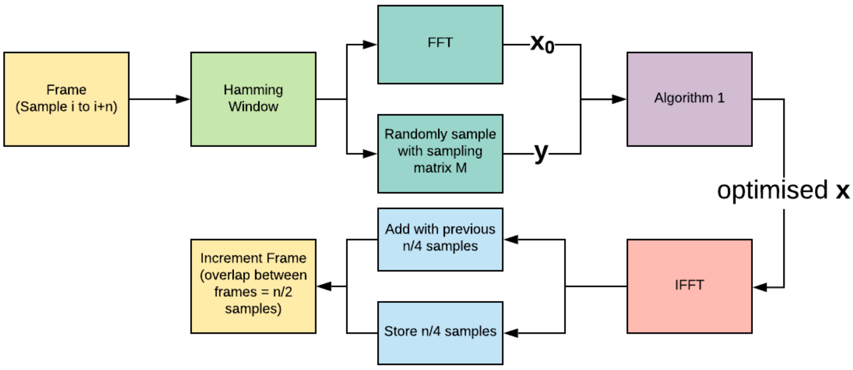


Figure 1: Implementation Flowchart

```
  res = vec - y
  vec = np.zeros(x.shape, dtype=np.complex128)
  vec[ii] = res
  vec = np.fft.fft(vec)/n
  return lm*(2*vec - (2/n/snr)*x) + shrink(x)

def grad_f_lm(x, y, ii):
  # Gradient of objective wrt Lagrange multiplier
  xt = np.fft.ifft(x)
  vec = xt[ii]
  T1 = np.sum((np.abs(vec- y))**2)
  T2 = np.sum(np.abs(x)**2)/n/snr
  return (T1 - T2)
```

5 Results

Audio clips from the TIMIT[7] dataset were chosen to run tests on them. Different synthetic noise samples were added to the audio signals to test out the algorithm. The noises that were tested are primarily Stationary White Gaussian Noise (SWGN), Non-Stationary White Gaussian Noise (NSWGN) and random Cauchy Noise (digital noise sampled from a stationary Cauchy distribution). In all the following places, input SNR is considered as the ratio of average signal power to average noise power.

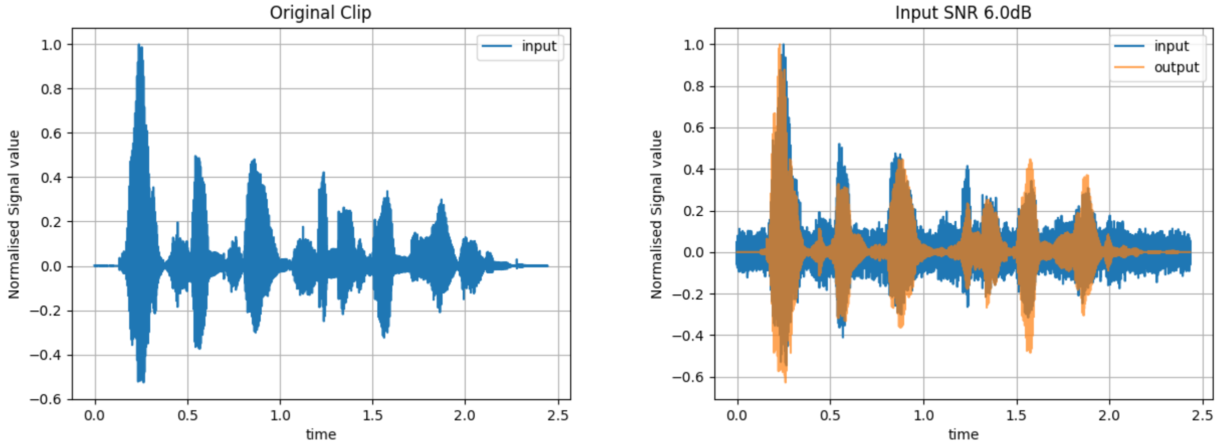
On running the algorithm we find that the major drawback of this algorithm is that the voice is made to sound robotic. This is an artefact of the algorithm as it tries to find the best sparse representation of the audio clip and in turn loses some of the more intricate details of human speech.

In figure 2, we see from the audio waveforms that the algorithm has managed to remove the additive white noise quite well.

In figure 3, we see that the denoising algorithm is mildly robust to additive digital Cauchy noise.
For the Test-2 audio clip, non-stationary white Gaussian noise was added and we see from figure 4 that the algorithm works well with it too.
Another side effect of the algorithm is a compressed representation of the audio signal which can be exploited for other applications. On checking the output frequency vectors contained a maximum of 64 non zero elements, which shows tells us that there is a compression of about $1024/64 = 32$ for each frame. However, information about speech is lost.
The source code along with the audio test cases and results can be found here.

6 Further Scope

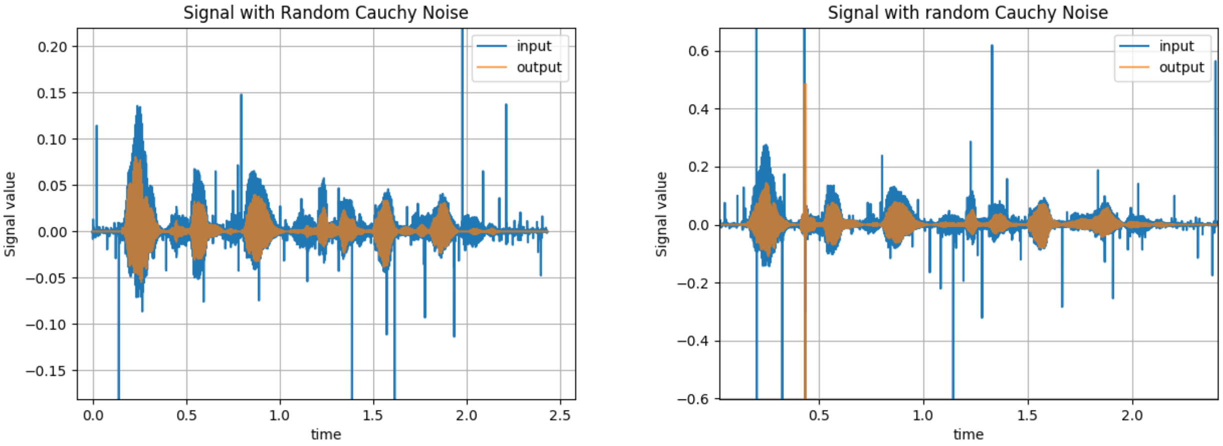
- One of the major disadvantages of using this method is that the sparsity reduces the “quality” and the feel of human-like voice. We can try mitigating it by estimating higher harmonic contents in the frame after the sparsity constraint.
- One major advantage of this method is the $\sim 32\times$ compression factor achieved. One can try to exploit this for robust coding and denoising of voice signals.



(a) Original Clip without noise

(b) Output of algorithm with added SWGN

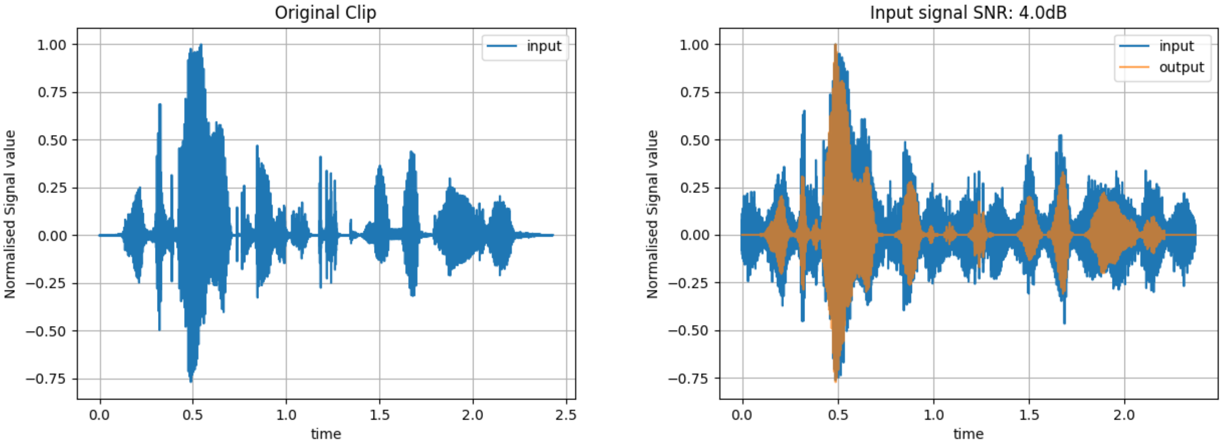
Figure 2: Results with Test-1 Audio clip



(a) Average SNR of input: 6.0dB

(a) Average SNR of input: -3.0dB

Figure 3: Results with Test-1 and added Cauchy Noise



(a) Original Clip without noise

(b) Output of algorithm with added NSWGN

Figure 4: Results with Test-2 Audio clip

References

[1] Candes, E. J., Romberg, 1. and Tao, T. *Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information*. IEEE Transactions on Information Theory (Volume: 52, Issue: 2,

Feb. 2006)

[2] Dalei Wu, Wei-Ping Zhu and M.N.S. Swamy. *A Compressive Sensing Method for Noise Reduction of Speech and Audio Signals*. 2011 IEEE 54th International Midwest Symposium on Circuits and Systems (MWSCAS).

[3] Jacob Mattingley and Stephen Boyd. *Real-Time Convex Optimization in Signal Processing*
https://web.stanford.edu/~boyd/papers/pdf/sig_proc_mag.pdf

[4] *Lecture Slides on Primal-Dual Subgradient Method - Stanford University - EE364b Convex Optimisation*
https://web.stanford.edu/class/ee364b/lectures/primal_dual_subgrad_slides.pdf

[5] Kai Xiong, Guanghui Zhao, Guangming Shi, and Yingbin Wang. *Convex Optimization Algorithm for Compressed Sensing in a Complex Domain: The Complex-Valued Split Bregman Method*
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6832202/

[6] Ryan Tibshirani. *Convex Optimization - Lec 6: Sub-gradients*
https://www.stat.cmu.edu/~ryantibs/convexopt-F15/lectures/06-subgradients.pdf

[7] *TIMIT-corpus voice dataset*
https://www.kaggle.com/nltkdata/timitcorpus/data?select=timit!