Time Frequency Weighted Overlapping Group Shrinkage for Speech Denoising

Alex Epstein, Nimish Magre
Northeastern University, Boston, MA
epstea@amazon.com, magre.n@northeastern.edu

Abstract

Developing on the idea of using the predetermined structural knowledge in convex optimization problems, we focus specifically on the task of denoising speech samples. To this end, we exploit the grouping/clustering property observed with speech spectograms to iteratively obtain a sparse clean speech signal using a mixed norm penalty term. We build upon the Overlap Group Shrinkage (OGS) algorithm [3] and introduce time-frequency weighting to the cost function to rid the sparse clean signal of the residual noise. This time-frequency weighting also empirically is shown to extend the algorithm to effectively handle impulsive noise types whereas the original algorithm was geared towards white gaussian noise. We also notice that our extensions for both time & frequency are generic meaning that we can adapt our frequency weighting for the specific type of target signal we are attempting to extract or from the inverse perspective the noise we are trying to suppress. Additionally depending on how the target or noise varies over time we are able to focus on certain time slices more heavily than others and that further improves the performance of the algorithm in terms of both the SNR and intelligibility.

1. Introduction

Unsupervised speech denoising, especially to eliminate musical noise has been been an important research domain recently. The problem involves decomposing a noisy speech sample *Y* into its clean speech component *X* and the noise component *N*. A lot of the research conducted in the domain tends to sparsify the clean speech component *X* using convex optimization methods with an l_1*-norm* penalty term to promote sparsity [12][2].
However, the l_1*-norm* penalty term implicitly assumes the signal coefficients to be independent and therefore fails to effectively eliminate the residual noise from the sparse clean speech samples[1]. As with wavelet coefficients for physically arising signals, speech signals also tend to display the clustering property showing inter/intra-scale clustering (i.e signficant values of *X* tend to not be isolated). This pre-known structural property could be exploited as done by [3] to further improve the SNR results of the clean speech by using a group-sparsity promoting penalty term in the cost function. Through further experimentation, we observed that while the introduction of the $l_1, 2$*-norm* penalty term helped obtain higher SNR scores, the metric did not represent the clean speech intelligibility accurately. It was also observed that speech sparsification through this algorithm provided poor results when subjected to impulsive noise. since the optimization process was conducted in the Fourier domain and the denoising was conducted specifically on speech signals, we further extended the algorithm by introducing scaling factors to the cost function based on time slices that make use of the average energy for all frequencies in every time-slice and frequency bins that limit the extracted signal to a typical speech frequency range. This modification to the cost function not only helped the algorithm in adapting better to impulsive noise suppression but also made the algorithm generic (i.e the algorithm was fit to suppress multiple noise-types as long as their type was known apriori).

2. Related Work

2.1. MM Algorithms[4]

Matrix decomposition problems with sparsification constraints have been popularly reformed into convex optimization problems solved iteratively through thresholding algorithms such as the Alternating Direction Method of Multipliers (ADMMs) [12][2]. However in order to iteratively optimize over the objective function, these algorithms tend to make use of variable splitting and therefore increase the computational cost in terms of memory and speed.
The Majorize-Minimization or Minorize-Maximization optimization method is an alternative to such thresholding algorithms. The MM algorithms consider optimizing over an easier surrogate function in order to optimize over the required objective function.
Consider the concave objective function $f(\theta)$ to be maximized in Figure 1:

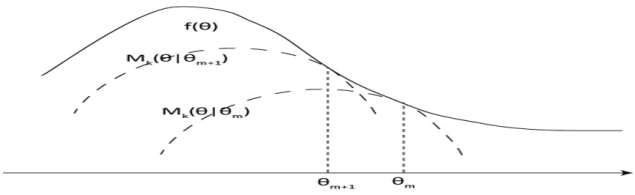

Figure 1. Minorize-Maximization method to maximize the concave objective function.[6] using a quadratic surrogate function

To maximize over this function, a surrogate quadratic function (minorized version of the objective function) is chosen. At step m of the optimizing algorithm, this quadratic function $g(\theta|\theta_m)$ satisfies two particular properties required when choosing the surrogate function:

1. **Tangency:** $g(\theta|\theta_m) = f(\theta_m)$ i.e. at θ_m, the surrogate function is tangential to the objective function
2. **Dominance:** $g(\theta|\theta_m) \leq f(\theta)$ for all θ i.e. the objective function is always greater than or equal to the surrogate function at step m

The algorithm then maximizes over the surrogate function to obtain the following update step:

$$\theta_{m+1} = \arg\max_{\theta} g(\theta|\theta_m) \tag{1}$$

Then through the following construction, it can be said that this iterative method will lead to the objective function $f(\theta)$ to converge as m goes to infinity:

$$f(\theta_{m+1}) \geq g(\theta_{m+1}|\theta_m) \geq g(\theta_m|\theta_m) = f(\theta_m) \tag{2}$$

A similar approach is taken in the Majorize-minimization algorithm but the objective function to be minimized is convex.

2.2. Overlap Group Shrinkage (OGS)[3]

For applications involving real-world signals, such as classification, compression and synthesis, the wavelet domain has been a natural setting. Often, each significant coefficient (i.e a wavelet, a coefficient of the Short Term Fourier Transform or a large signal amplitude) is treated as being independent of all others and these coefficients are modelled either as jointly Gaussian or as non-Gaussian but independent [10]. However, with real natural signals, neighbouring coefficients tend to have statistical dependency and follow the clustering property which indicates that if a particular coefficient is large/small in amplitude, then the adjacent coefficients also tend follow similar behavior [8][11]. This intra/interscale grouping property is similarly observed within typical speech spectrograms.
This paper addresses the problem of denoising these real-world signals which display the clustering property through estimating a sparse signal (x(i), $i \epsilon I$) from noisy observations (y(i)). Algorithms that utilize hard/soft thresholding functions to obtain the sparse solutions typically tend to utilize the l_1-*norm* to induce sparsity and overlook the grouping property [12][2]. To obtain the sparse clean signal (x(i)) from the noisy observations (y(i)), the optimization problem could be generally formulated as:

$$x^* = \arg\min_x F(x) = \frac{1}{2}||y - x||_2^2 + \lambda R(x) \tag{3}$$

where the penalty function (R(x)) is chosen to promote the group sparsity behaviour of signal x. The proposed OGS algorithm in this paper minimizes the cost function represented by Equation 3 with the non-separable penalty term:

$$R(x) = \sum_{i \in \mathcal{I}} [\sum_{j \in \mathcal{J}} |x(i+j)^2]^{\frac{1}{2}} \tag{4}$$

where the set J defines the group. From a Bayesian perspective, the l_1-*norm* implicitly assumes the signal coefficients to be independent; however, the penalty term Equation 4 introduced by Ilker Bayram [1] for signal denoising makes use of mixed norms to induce the group sparsity property. Unlike other shrinkage/thresholding algorithms such as ADMMs that require the use of variable splitting for the iterative solving of Equation 3, the proposed OGS algorithm makes use of the Majorization-Minimization (MM) [4] method wherein Equation 3 is solved via the introduction of an easier to optimize surrogate function while the algorithm itself uses separable convolutions to calculate the penalty term. This method is therefore computationally less expensive in terms of speed and memory compared to variable splitting based thresholding algorithms.
Since J represents the group size in Equation 4, and a sliding window is shifted over each sample at a time, it is apparent that the penalty function is translation-invariant i.e the groups are fully overlapping. In iteratively solving Equation 3, the MM method promotes majorization of the penalty term while minimizing the cost function and produces the sequence:

$$x^{(k+1)}(i) = \arg\min_{(x \in \mathcal{C})} (y(i) - x)^2 + \lambda r(i; x^{(k)})|x|^2 \tag{5}$$

to be iteratively updated by

$$x^{(k+1)}(i) = \begin{cases} \frac{y(i)}{1+\lambda r(i;x^{(k)})}, & i \in \mathcal{I}', \\ 0 & otherwise. \end{cases} \tag{6}$$

This iteration step defines the OGS algorithm and is better explained below in the form of pseudo-code:

Algorithm 1 The OGS Algorithm

input: $\mathbf{y} \in \mathbb{R}, \lambda > 0, \mathcal{J}$

x=y (initialization)

$\mathcal{I}'$ = $\{\mathrm{i} \in \mathcal{I}, x(i) \neq 0\}$

repeat

$r(i;x)$= $\sum_{j\in\mathcal{J}} \sum_{k\in\mathcal{J}} |x(i-j+k)|^2]^{-\frac{1}{2}}, i \in \mathcal{I}'$

$x(i) = \frac{y(i)}{1+\lambda r(i;x)}, i \in \mathcal{I}'$

until convergence

return : x

Important to note in the above algorithm is that if x is initialized to zero, then all the consecutive updates will lead to an optimized x equal to zero due to the penalty term r(i;x) being undefined. Instead of utilizing direct thresholding to bring non-zero values of y(i) to zero, the OGS algorithm through its iterative step, gradually reduces these values towards zero thus promoting sparsity efficiently.

2.3. Speech Denoising via Low-Rank and Sparse Matrix Decomposition[5]

This denoising method follows the conventional separation of a noisy speech spectrogram into a low rank noise structure and a sparse clean speech structure with the addition of a residual noise term to rid the denoised speech of the musical noise. The authors therefore propose to separate the noisy speech structure *Y* into three submatrices: a low rank noise structure (*L*), a sparse speech structure (*S*) and a random residual noise structure (*R*). The random residual noise structure *R* is assumed to follow independent and identically distributed zero-mean Gaussian distributions.
Based on the structural assumptions made regarding *L*, *S* and *R*, the optimization problem is formulated as

$$Y = L + S + R, rank(L) \leq r \; and \; card(S) \leq k \quad (7)$$

To solve the above problem iteratively, it is reformulated as:

$$\min_{L,S} ||Y - L - S||_F^2 + \lambda ||S||_1, \; s.t \; \; rank(L) \leq r \quad (8)$$

since estimating the cardinality of *S* is practically difficult. Important to note is also the fact that the low-rank *L* and sparse *S* are found iteratively while being perturbed by the residual noise *R*. Following the solutions for *L* and *S* found using the Semi-Soft GoDec algorithm [13], the authors further apply a binary mask [7] to the noisy speech spectrogram (*Y*) using

$$\hat{S}(l,n) = M(l,n).Y(l,n) \quad (9)$$

The binary mask applied here is decided using

$$M(l,n) = \begin{cases} 1, & |S(l,n)| > |L(l,n)|, \\ 0 & otherwise. \end{cases} \quad (10)$$

and is estimated based on the ideal *S* and *L* structures found using the Semi-Soft GoDec algorithm. Finally the denoised speech spectrogram is transformed back to the time-domain by applying the inverse STFT function.

3. Method

3.1. Frequency Weighting

The first extension that we make to the OGS algorithm is to specialize it for extracting signals that generally sit within a typical frequency range. In our examples we have utilized a lowpass filter up to 4 kHz where we utilize the gain on this filter as a scaling factor for the importance of OGS to adhere to this formulated filter. The implemented filter's magnitude and phase response can be seen in Figure 2 which was derived using Parks-McClellan's algorithm for linear phase filter design. While we specialized our filter for speech intelligibility this concept could easily be extended to many other types of signals. We try to eliminate many of the ripples that are present in the Fourier domain by taking our filter in this domain and smoothing it. Without doing so horizontal lines of energy will be present in our denoised STFT at the same frequency where these ripples peak. When running OGS we provide an STFT representation of the signal that utilizes an FFT length of 512 and so we can take any set of filter coefficients and transform them to the Fourier domain which exactly provides 512 coefficients that we can multiply against a single time slice in our STFT element wise. The math for the explanation above can be found in subsection 3.3 This concept can also be utilized not just if our desired signal sits within a certain range, but if the interfering signal also sits in a certain range we can use a band stop filter to suppress this noise, however this requires apriori knowledge of what the noise looks like. Generating the frequency weights is completely invariant to the data size and is a fixed parameter based on the FFT length that we have chosen. Meaning there is no associated cost that scales with the size of the data, making the compute & memory impact here constant and minimal.

3.2. Time Weighting

The next addition we made to the OGS algorithm was to vary the attenuation across time similar to what we did across frequency. This addition is what heavily improved our ability to handle impulsive noise types in a way that the original OGS algorithm could not. The intuition here is that if our speech signal is generally at a consistent energy level then what should be varying the most is the noise.

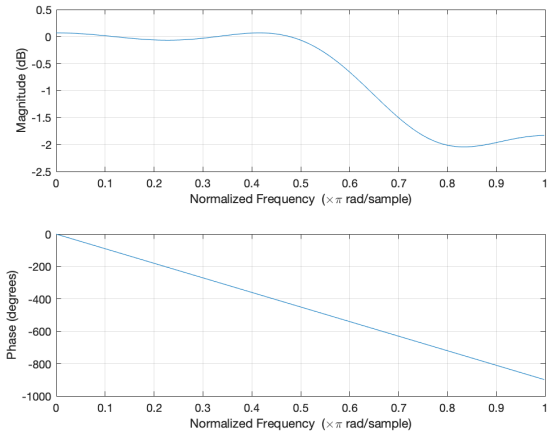

Figure 2. Designed lowpass filter for speech extraction before applying any gain or smoothing.

While there are many possibilities for varying the attenuation attention across time our implemented method is as follows: starting with the STFT representation of our noisy signal take the squared sum across the frequency bins for each time slice to end up with a row vector, simply take the element wise division of this row vector compared to the average value of this row vector and then apply some scaling to this result that corresponds to the importance of OGS adhering to these weights. The math for the explanation above can be seen in subsection 3.3. In essence what this does is have OGS focus on attenuating time slices with energy that is larger than the average energy for any given time slice. Creating these time weights is a one-time cost run before the iterative steps of the algorithm, however it does scale with the size of the data linearly as we will have more time bins to create weights for the longer the audio is that we are trying to denoise.

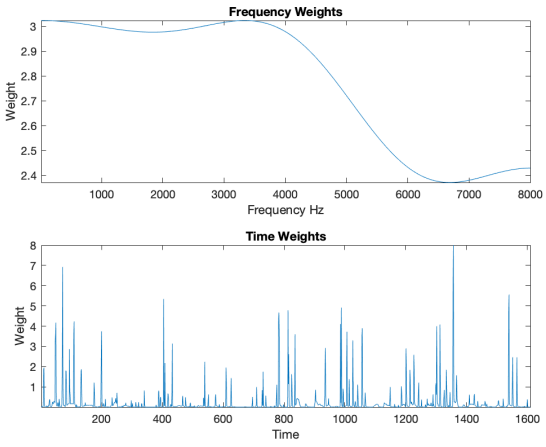

Figure 3. Example frequency weighting with gain and smoothing applied along with time weighting.

3.3. Cost Function Modification

In the following equations y is the STFT representation of our original noisy signal x is the STFT representation of our target signal while x^* is the approximation of our target signal. We will use f_l to represent the FFT length and t_l to represent the number of time slice bins in our STFT representation. Additionally we will use t_s for the time gain/scaling factor and f_s for the frequency gain/scaling factor.
The modified cost is as follows:

$$x^* = \arg\min_x F(x) = \frac{1}{2}||y - x||_2^2 + W(y) \odot R(x) \quad (11)$$

where

$$W(y) = \mathbf{1}_{f_l \text{x} 1} * W_t \odot \mathbf{1}_{1 \text{x} t_l} * W_f \quad (12)$$

In Equation 12 we are creating our weight matrix from a frequency weighted column vector and a time weighted row vector through typical matrix multiplication ($*$). Then we take these two matrices that weight time and frequency respectively and perform element wise multiplication ($\odot$) between them to get our final weight matrix.

$$E_t = [\sum_{i=1}^{f_l} |y_{i*}|^2]_{1 \text{x} t_l} \quad (13)$$

In Equation 13 we are calculating the average power for each time slice across all the frequency bins.

$$E_r = [\frac{E_t}{Avg(E_t)}]_{1 \text{x} t_l} \quad (14)$$

In Equation 14 we divide this row vector by the average value of this vector to see how the energy in each time slice compares to the average energy of the signal.

$$W_t = [t_s \frac{E_r}{Max(E_r)}]_{1 \text{x} t_l} \quad (15)$$

Finally in Equation 15 we divide by the maximum value from Equation 14 to bound our values between 0 & 1 and then we apply our scaling factor t_s.

$$W_f = [f_s |\mathcal{F}[H(z)]|]_{f_l \text{x} 1} \quad (16)$$

Lastly in Equation 16 we create our column vector of weights by taking our filter coefficients $H(z)$ and utilize a Fourier transform to bring them to the frequency domain. We then take the magnitude of these weights and apply the scaling factor f_s to get our final frequency column vector.
With these modifications to the cost function to account for both the time and frequency weighting we have not theoretically proven the monotonically decreasing behavior as seen in the original OGS paper. However empirically this behavior seems to be preserved by these modifications. The psuedocode for the modified implementation can be seen in Algorithm 2.

Algorithm 2 The TFOGS Algorithm

input: $\mathbf{y} \in \mathbb{R}, H(z), f_s, t_s, \mathcal{J}$

x=y $(initialization)$

$\mathcal{I}'$ = $\{\mathrm{i} \in \mathcal{I}, x(i) \neq 0\}$

$\mathrm{W}_f = f_s * |\mathcal{F}[H(z)]|_{f_l \mathrm{x} 1}$

$E_t = [\sum_{i=1}^{f_l} |y_{i*}|^2]_{1\mathrm{x}t_l}$

$E_r = [\frac{E_t}{Avg(E_t)}]_{1\mathrm{x}t_l}$

$W_t = [t_s \frac{E_r}{Max(E_r)}]_{1\mathrm{x}t_l}$

$W = \mathbf{1}_{f_l \mathrm{x} 1} * W_t \odot \mathbf{1}_{1\mathrm{x}t_l} * W_f$

repeat

$r(i;x) = \sum_{j \in \mathcal{J}} [\sum_{k \in \mathcal{J}} |x(i-j+k)|^2]^{-\frac{1}{2}}, i \in \mathcal{I}'$

$x(i) = \frac{y(i)}{1+W \odot r(i;x)}, i \in \mathcal{I}'$

until *convergence*

return: x

3.4. Low-Rank Noise Structure with Binary Masking

Following from the findings by Huang J. et al [7], we attempt to obtain the low-rank noise structure from the TFOGS algorithm by performing Singular Value Thresholding on the output noise structure for the case with White Gaussian Noise applied on the clean signal. To this end, we approximate the low-rank structure of the noise matrix by considering the top 5, 50 and 100 singular values.

Following this low-rank approximation of the noise structure and the sparse clean speech obtained from the TFOGS algorithm, we apply Equation 9 and Equation 10 to perform binary time-frequency masking on the noisy signal (y(i)) in the STFT domain to further suppress the 'musical' residual noise.

Upon performing the binary-masking using the low-rank noise structure and sparse clean speech structure, we obtained poor SNR values of 1.91, 1.97 and 2.04 when considering the top 5, 50 and 100 singular values for the noise structure respectively. Even in terms of intelligibility, the binary masking failed to effectively suppress the musical noise from the noisy signal *Y*. We believe that one of the reasons for this observed failure is the hard 1-0 masking conditions of the binary mask. We intend to further experiment with adaptive masking in place of binary masking to improve on our results. Adaptive masks allow for masking conditions in the range 0-1 instead of hard 0-1 masking conditions and could therefore help in regulating the sparsified instances.

4. Examples and Results

For our experiments we dealt with two different types of noise. Similar to the original OGS paper we dealt with White Gaussian Noise to show that our extensions provided similar or better performance. Additionally we showed how our new extensions equipped the OGS algorithm to handle impulsive noise that was not present throughout the whole signal like WGN, but heavily decreased overall intelligibility of the speech.

4.1. White Gaussian Noise

In the WGN case it still seems that our extension has not improved the algorithm by the objective SNR metric. In terms of subjective listening for intelligibility we have similar performance however this can suffer from researcher/confirmation bias as we were unable to collect a randomized sample of individuals to compare the two output samples. In this case since WGN has equal power across all frequencies we actually set our frequency weighting to a matrix of one's essentially removing this weighting. However we did keep the time weighting based on the energy scaling as described before and while we don't notice any major differences here because the energy is very evenly dispersed across time as well, theoretically this should attenuate time slices with a large noise energy to at least make the energy throughout the denoised signal a bit more constant.

4.2. Impulsive Noise

Empirically we have noticed that our extensions on top of OGS have the largest impact for impulsive noise types. This is due to the fact that a spike in energy within a time slice corresponds very strongly to noise and our time weighting attenuates this time slice more than ones with less energy. While our frequency weighting ensures that we preserve the band of importance for intelligibility so we have a nice balancing act between attenuating very noisy sections of our signal in time while ensuring we do not remove important sections of our signal in frequency. We can see the comparison between the last iteration of our modified OGS algorithm in Figure 4 versus the original algorithm in Figure 5. We can see the importance that the time and frequency weighting places upon the update step resulting in sparsity only for given frequency bands rather than the entire time slice as seen in the original algorithm. This modified type of sparsity for a target speech signal leaves it to be much more intelligible than it's original counterpart. Unfortunately with the original OGS algorithm if we try to remove less noise to decrease sparsity overall then the SNR is so low that it is still left as unintelligible. We can see how these effects show up in the time domain for the original OGS algorithm in Figure 7 and the TFOGS algorithm in Figure 6. Additionally in these figures we can see the

Table 1. Comparing different noise types for the original vs the modified OGS algorithm.

Noise Type	SNR (dB)	OGS SNR (dB)	TFOGS SNR (dB)
White Gaussian	-0.87	8.21	7.87
Impulsive	-7.42	-3.8	4.43

weighting matrix along with few other spectrograms that help understand the performance of each algorithm.

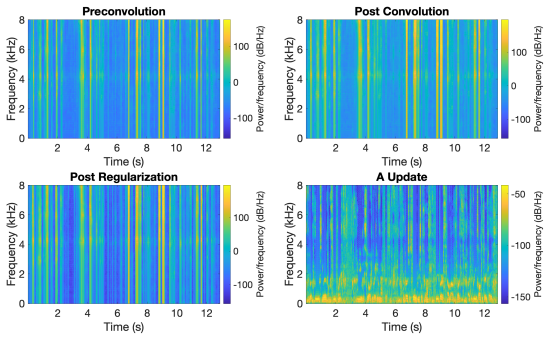

Figure 4. Last iteration intermediate values from time frequency weighting approach for impulsive noise.

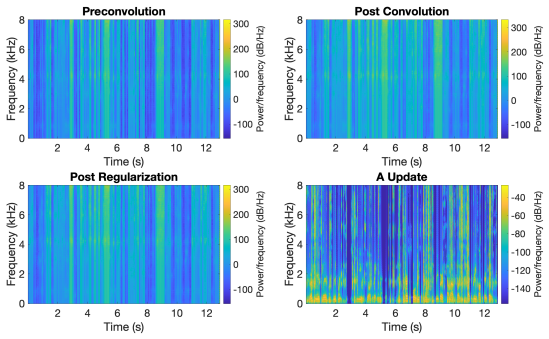

Figure 5. Last iteration intermediate values from original regularization approach for impulsive noise.

4.3. Metrics

When it comes to speech it is not reliable to utilize SNR as the only metric for determining how well the algorithm worked. In the intuitive sense, you can greatly increase the SNR but at the same time sacrifice energy in specific frequency bins that are needed for humans to be able to distinguish between different phonemes. So while the parameters we have chosen for our examples are not the best by any objective metric that we calculated we empirically played a balancing act between intelligibility and SNR to describe the performance of the algorithms. There is a lot of work in regards to finding an objective metric that can be used for speech intelligibility [9]. However for the sake of having objective metric measurements we have our SNR results in Table 1.

5. Conclusion

In this work we proposed an extended version of OGS specifically targeted for audio that given some apriori knowledge either about our noise or target signal can enhance the performance of the algorithm. We focused on having target signals of speech and removing both Gaussian and impulsive noise, however this could be further extended to work with other noise types or target signals. The framework we have proposed here can be generalized and further extended to modify how the frequency or time weights are created along with the various parameters such as the scaling factors. We have found in our experimentation that these additions help remove more noise from the corrupted signals while keeping relevant information that increases our speech intelligibility. We did try and continue to extend the algorithm by imposing low rank constraints on the noise and then utilizing binary masking to further increase SNR & intelligibility unsuccessfully, but we do believe this path can provide fruitful results given the necessary time. Furthermore it would be advantageous if this work could be further extended to automatically determine some of the parameters we have introduced here. This would make it so the algorithm can be run blindly without hand tuning parameters and making it easier on users of the algorithm. All the additions we have made have very minimal compute or memory requirements keeping the TFOGS algorithm essentially as lightweight as the original OGS algorithm.

6. Contributions

Both the authors provided equal contribution in performing literature review, preparing the project proposal, preparing the project presentation and writing the final report. Alex was responsible for implementing the infrastructure code, generating the necessary graphs to analyze the results and implemented the frequency weighting method 3.1. Nimish performed the experimentation with using binary masking 3.4 and implemented the time weighting method 3.2 using infrastructure code implemented by Alex.

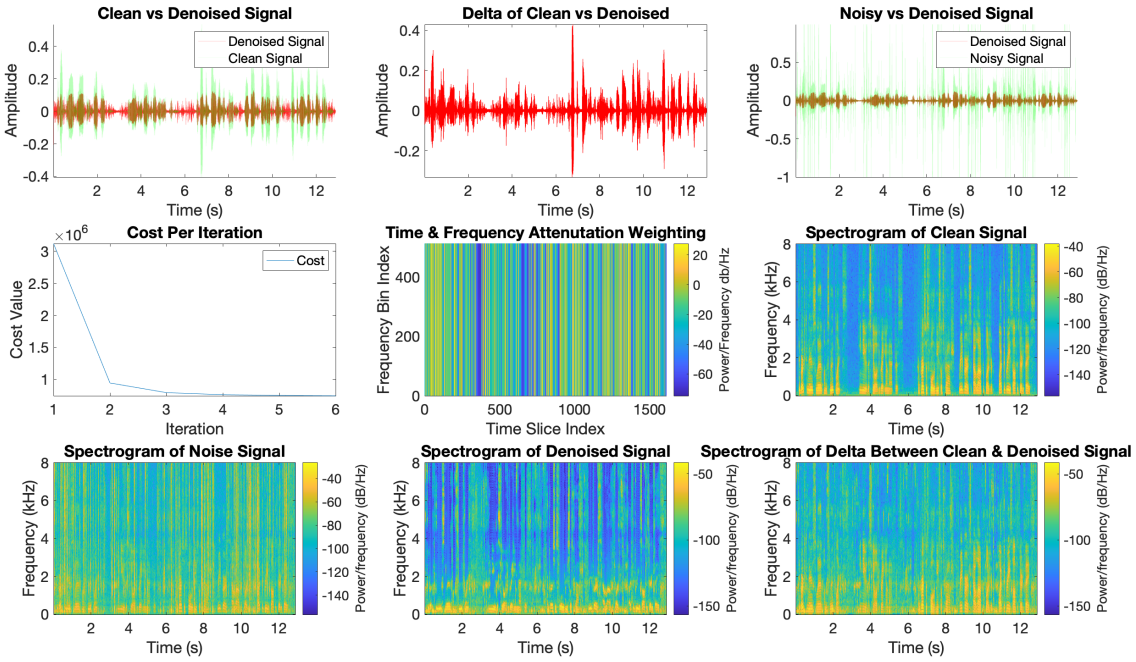

Figure 6. Output analysis from time frequency weighting approach for impulsive noise.

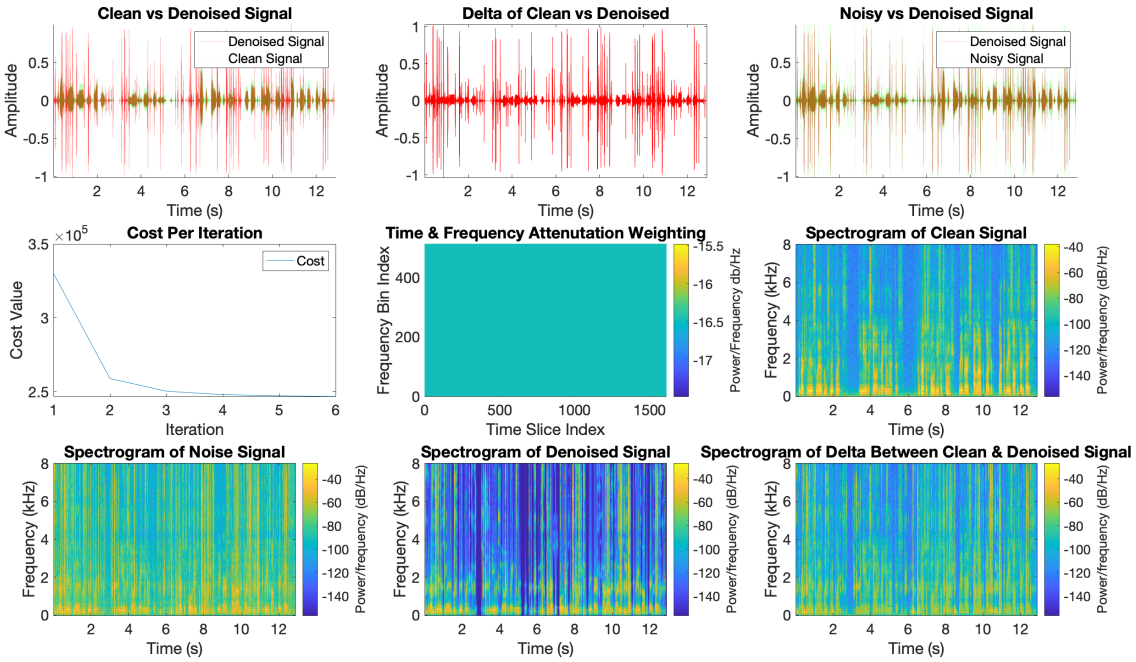

Figure 7. Output analysis from original regularization approach for impulsive noise.

References

[1] Ilker Bayram. "Mixed norms with overlapping groups as signal priors". In: *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2011, pp. 4036–4039.

[2] Scott Shaobing Chen, David L Donoho, and Michael A Saunders. "Atomic decomposition by basis pursuit". In: *SIAM review* 43.1 (2001), pp. 129–159.

[3] Po-Yu Chen and Ivan W Selesnick. "Translation-invariant shrinkage/thresholding of group sparse signals". In: *Signal Processing* 94 (2014), pp. 476–489.

[4] Mário AT Figueiredo, José M Bioucas-Dias, and Robert D Nowak. "Majorization–minimization algorithms for wavelet-based image restoration". In: *IEEE Transactions on Image processing* 16.12 (2007), pp. 2980–2991.

[5] Jianjun Huang et al. "Speech denoising via low-rank and sparse matrix decomposition". In: *ETRI Journal* 36.1 (2014), pp. 167–170.

[6] Kenneth Lange. *The MM Algorithm*. Apr. 2007.

[7] Yipeng Li and DeLiang Wang. "Musical sound separation based on binary time-frequency masking". In: *EURASIP Journal on Audio, Speech, and Music Processing* 2009 (2009), pp. 1–10.

[8] Juan Liu and Pierre Moulin. "Information-theoretic analysis of interscale and intrascale dependencies between image wavelet coefficients". In: *IEEE Transactions on Image processing* 10.11 (2001), pp. 1647–1658.

[9] Jianfen Ma, Yi Hu, and Philipos C. Loizou. "Objective measures for predicting speech intelligibility in noisy conditions based on new band-importance functions". In: *The Journal of the Acoustical Society of America* 125.5 (May 2009), p. 3387.

[10] Eero P Simoncelli and Bruno A Olshausen. "Natural image statistics and neural representation". In: *Annual review of neuroscience* 24.1 (2001), pp. 1193–1216.

[11] Eero P Simoncelli and Bruno A Olshausen. "Natural image statistics and neural representation". In: *Annual review of neuroscience* 24.1 (2001), pp. 1193–1216.

[12] Robert Tibshirani. "Regression shrinkage and selection via the lasso". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 58.1 (1996), pp. 267–288.

[13] Tianyi Zhou and Dacheng Tao. "Godec: Randomized low-rank & sparse matrix decomposition in noisy case". In: *Proceedings of the 28th International Conference on Machine Learning, ICML 2011*. 2011.