Dartmouth Research Experience

Variations of the MMV Problem

Mariah Boudreau[1] Carley Walker[2]

[1]Saint Michael's College, Colchester, VT

[2]University of Southern Mississippi, Hattiesburg, MS

August 8, 2018

Outline

Overview of Problem

Given a blurry image, or noisy signal, one can "deblurr" the signal to get a clearer picture.

- Signal: B (a measurement vector)
- Transform Matrix: A
- "True Image": f

$Af = B$ is the equation to solve in order to get the true image.

Now, since we are constructing our own data, we can assume error in our true equation, giving us

$$Af = B + \varepsilon.$$

Since A is an ill-posed matrix, we minimize f in order to generate estimates of the true signal.

$$\arg\min_{f} ||Af - B||_2^2 + \lambda||\mathcal{L}f||_1$$

Outline

Fourier Coefficients Reconstruction

Mariah Boudreau
Saint Michael's College

Example

Imagine this piece-wise function for the interval [-π, π] :

$$f(x) = \begin{cases} 1 & -\pi \leq x \leq \pi \\ 0 & \textit{otherwise} \end{cases}$$

We want to estimate this function.

Data had to be created by using a Fourier coefficients to simulate a B measurement vector.

$$B = \frac{1}{2\pi}\int_{-\pi}^{\pi} f(x)e^{-ikx}dx$$

$$\text{for } k = -N \ldots N$$

Fourier Coefficients Reconstruction

Continued

The largest k value determines the largest Fourier coefficient. After constructing data from $[-N, N]$, we can use the Discrete Fourier Transform to construct the transform matrix A. Since we are switching from continuous to discrete, we already know there is going to be a certain amount of error in our reconstruction.

The Discrete Fourier Transform Matrix is constructed using the equation below, where $n = 2N + 1$ and x_j is the data point at the j^{th} point for $j = 1 ... n$. Also, $k = -N ... N$:

$$A = \frac{1}{2n} f(x_j) e^{-ikx_j}$$

For j = 1 and j = n, those terms are $\frac{1}{2}(\frac{1}{2n} f(x_j) e^{-ikx_j})$

This constructs an *nxn* matrix where n is equal to the total data points we want to reconstruct.

To reconstruct the true coefficients from the original B vector, we use the minimization:

$$\arg\min_f ||Af - B||_2^2 + \lambda ||f - f_{approx.}||_2$$

Where λ is a parameter and $f_{approx.}$ is the approximation of f calculated as $f_{approx.} = (e^{-ikx_j})B$ for $j = 1...n$ and $k = -N...N$

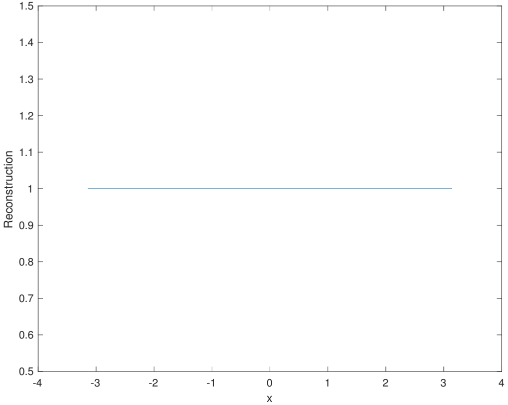

Figure: Reconstruction of the function from the B vector

We know that the Fourier coefficient vector reconstructs the function well, now we can add noise to multiple vectors. This simulates more data being collected (the multi-measurement vectors).
The minimization is the same as before but is done for each noisy data vector i.

$$\arg\min_{f^i} ||Af^i - B^i||_2^2 + \lambda||f^i - f^i_{approx.}||_2$$

Where λ is a parameter, and $f^i_{approx.}$ is the approximation of f^i calculated as $f^i_{approx.} = (e^{-ikx_j})B^i$ for $j = 1...n$ and $k = -N...N$

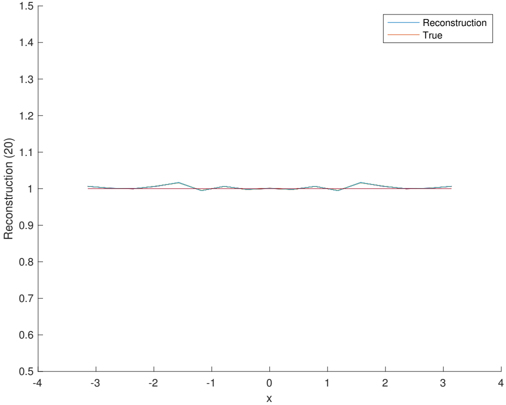

Figure: Reconstruction of the function from the 20 noisy data vectors

The optimal vector reconstruction out of all the data vector reconstructions can be chosen by using a Distance matrix:

$$Dist(i,g) = \sqrt{\sum_{k=1}^{n}(f_k^i - f_k^g)^2}$$

The optimal vector reconstruction is chosen by whichever row has the lowest sum in the Distance matrix.Then that reconstruction is the only one used from then on.

Fourier Coefficients Reconstruction

Continued

In order to adjust the optimal vector reconstruction using the other reconstructions, the variance (V) of each data point in the data vectors is calculated.
Utilizing the variance, weights for each data point in the optimal vector reconstruction can be applied by using this weight equation:

$$W_i = \frac{1}{V_i + \epsilon}$$

where $\epsilon = 1x10^{-12}$

These weights can be factored into the minimization problem. The new minimization is:

$$\arg\min_{f^{final}} ||Af^{final} - B^{opt.}||_2^2 + \lambda||f^{final} - f^{opt.}||_{1,W}$$

Where λ is a parameter

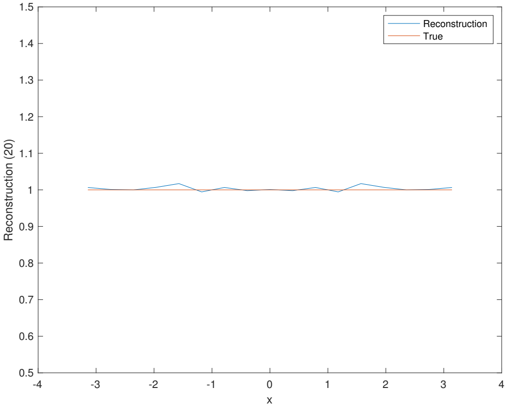

Figure: Reconstruction of the function from the optimal data vector with weighting

- More data vectors could be constructed
- More data points could be constructed
- The error could be adjusted to see how well the model does with more noise

Outline

Multiplicative Poisson Noise Reconstruction

Carley Walker
University of Southern Mississippi

Multiplicative Poisson Noise Reconstruction

Inconsistency in the deblurring model creates an additional model error on our inverse problem. For example, instead of our original inverse problem,

$$Af = B + \varepsilon,$$

we now look at

$$(A + C)f = B + \varepsilon$$

where C is an additive noise matrix supplement to the transform matrix. In this case, we will assume C is a **Poisson error distribution matrix** and the transform matrix A is a **discrete Fourier transform matrix**. We will also assume that our Poisson error distribution matrix has a **mean of zero**.

Obtaining our estimated data vectors will look like

$$\boldsymbol{y}^j = \left(A + C^j\right)\boldsymbol{f} + \boldsymbol{\eta}^j, \quad j = 1, \ldots, J$$

where $\boldsymbol{\eta}^j$ are the preexisting Gaussian noise vectors.

ℓ_1 regularization will provide an effective means for reconstruction of the true image given J noisy data vectors. We have the following optimization problem to find our MMV constructions using these estimated data vectors:

$$\hat{\boldsymbol{f}}^j = \arg\min_{\boldsymbol{g}} ||A\boldsymbol{g} - \boldsymbol{y}^j||_2^2 + \frac{\lambda}{2}||\mathcal{L}\boldsymbol{g}||_1, \quad j = 1, \ldots, J$$

where λ is the ℓ_1 regularization parameter and $\mathcal{L}$ is the 2^{nd} order polynomial annihilation transform.

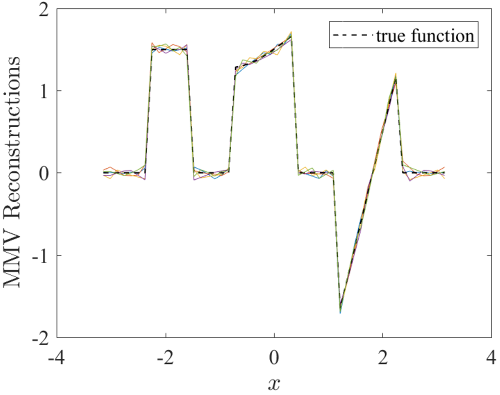

Figure: Five reconstructions, $\hat{\boldsymbol{f}}^j$, $j = 1, \ldots, 5$, of true image $\boldsymbol{f}$ with $\lambda = 0.3$, $n = 50$, $m = 50$, and $J = 10$.

Next, we must compute the variance of $\mathcal{L}\hat{\boldsymbol{f}}^j$, $j = 1, ..., J$, using

$$\hat{v}_i = \frac{1}{J}\sum_{j=1}^{J}\mathcal{P}_{i,j}^2 - \left(\frac{1}{J}\sum_{j=1}^{J}\mathcal{P}_{i,j}\right)^2, \quad i = 1, ...N$$

where $\mathcal{P}$ is the matrix of J vectors approximating some sparse feature of the underlying function $\boldsymbol{f}$, defined as

$$\mathcal{P} = [\ \mathcal{L}\hat{f}^1 \ \ \mathcal{L}\hat{f}^2 \ \ ... \ \ \mathcal{L}\hat{f}^J\ ], \in \mathbb{R}^{N\times J}.$$

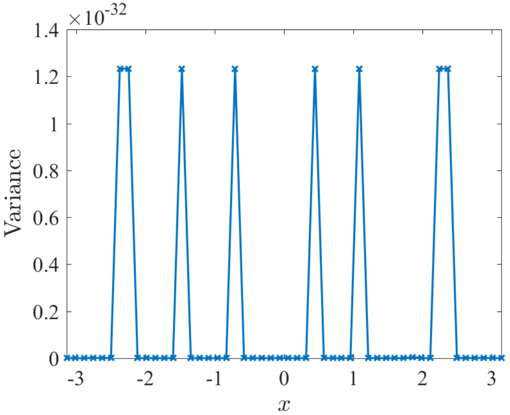

Figure: Variance graph of J reconstructions of true image $\boldsymbol{f}$ with $\lambda = 0.3$, $n = 50$, $m = 50$, and $J = 10$.

Next, from $\hat{v}_i$, we must determine the weights for the weighted ℓ_1 norm in the joint sparsity reconstruction. We calculate a vector of weights greater than zero:

$$w_i = \begin{cases} \mathscr{C}\left(1 - \frac{v_i}{max_i \quad v_i}\right), & i \notin I \\ \frac{1}{\mathscr{C}}\left(1 - \frac{v_i}{max_i \quad v_i}\right), & i \in I \end{cases}$$

where I consists of i such that $\frac{1}{J}\sum_{j=1}^{J} \tilde{\mathcal{P}}_{i,j} > \tau$. We choose τ so that when I is satisfied, we assume an edge at x_i, and that the index i is part of support in the sparsity domain of $\boldsymbol{f}$.

Normalizing the polynomial annihilation transform matrix gives us

$$\tilde{\mathcal{P}} = [\ \tilde{\mathcal{P}}_1\ \ \tilde{\mathcal{P}}_2\ \ \ldots\ \ \tilde{\mathcal{P}}_J\ ], \in \mathbb{R}^{N\times J}.$$

$\mathscr{C}$ we define as a **weighting scalar** that is the **average** ℓ_1 norm across all measurements of the normalized sparsifying transform of our measurements.

$$\mathscr{C} = \frac{1}{J}\sum_{j=1}^{J}\sum_{i=1}^{N}\tilde{\mathcal{P}}_{i,j}$$

This allows us to scale the weights according to the size of the values in the sparsity domain.

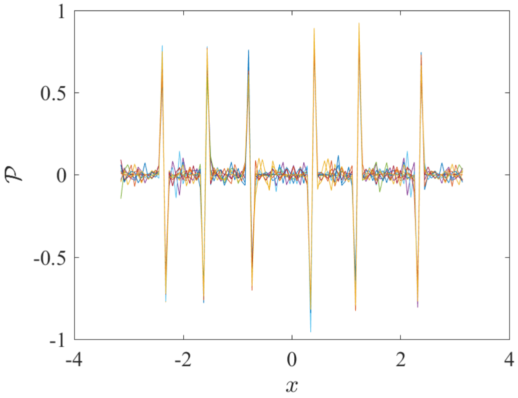

Figure: J corresponding sparsity vectors of our MMVs.

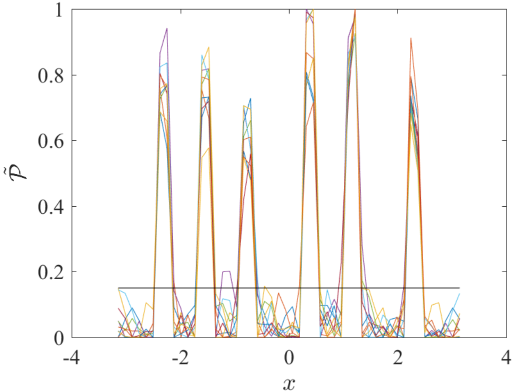

Figure: Illustrating the threshold of τ on the graph of $\tilde{\mathcal{P}}$ in order to detect edges and ignore noise. We choose a τ to be less than the average of the normalized PA transform. With our example, our MMVs' $\tilde{\mathcal{P}}$ means range from $0.1892 \leq \mu \leq 0.2366$, so we choose our τ to be 0.15.

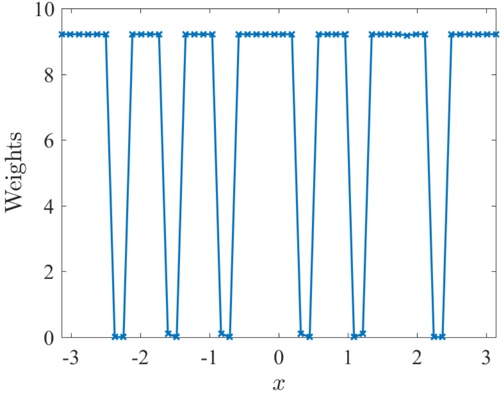

Figure: Weights graph of J reconstructions of true image $\boldsymbol{f}$ with $\lambda = 0.3$, $n = 50$, $m = 50$, $J = 10$, and a chosen τ of 0.15.

Next, we must find an optimal data vector $\hat{\boldsymbol{y}}$ that will enable us to avoid "bad" information in our joint sparsity reconstruction. We choose $\hat{\boldsymbol{y}}$ to be one whose corresponding measurements are nearest to the majority of the other J data vectors. We will define this to be the distance matrix $\mathcal{D}$, which will look like

$$\mathcal{D}_{i,j} = ||\hat{f}^i - \hat{f}^j||_2.$$

Our optimal data vector $\hat{\boldsymbol{y}}$ will correspond to the j^*th index that solves

$$(i^*, j^*) = \underset{1\leq i,j\leq J,\ \ i\neq j}{\arg\min}\ \mathcal{D}_{i,j}.$$

We are choosing the optimal column index j^* for our final reconstruction.

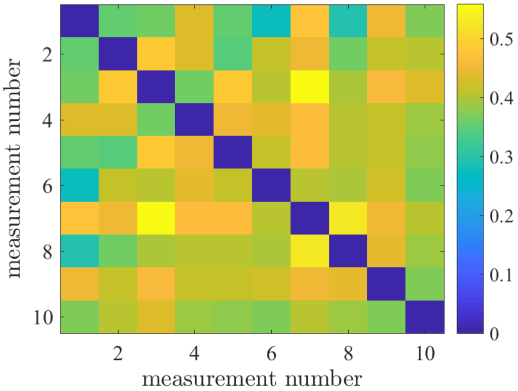

Figure: Construction of the distance matrix $\mathcal{D}$ with $\lambda = 0.3$, $n = 50$, $m = 50$, and $J = 10$.

Finally, using our newly obtained $\hat{\boldsymbol{y}}$, we solve the jointly sparse, weighted ℓ_1 minimization problem to conclusively reconstruct $\boldsymbol{f}$:

$$\hat{\boldsymbol{g}} = \operatorname*{arg\,min}_{\boldsymbol{g} \in \mathbb{R}^N} \ \frac{\lambda}{2} ||\mathcal{L}\boldsymbol{g}||_{1,w} + ||A\boldsymbol{g} - \hat{\boldsymbol{y}}||_2^2.$$

We are now *weighting* the ℓ_1 regularized polynomial annihilation transform for optimal edge detection, and we are minimizing $\boldsymbol{g}$ for $\hat{\boldsymbol{g}}$ with our optimal data vector.

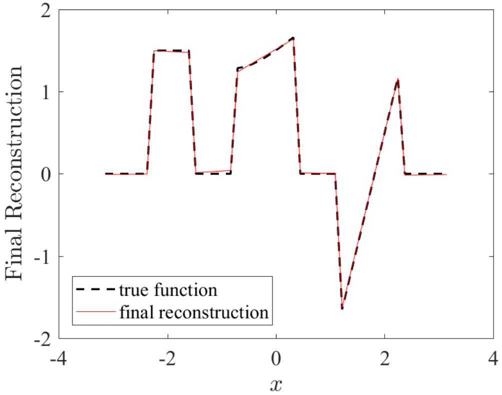

Figure: Final reconstruction $\hat{\boldsymbol{g}}$ of the true image $\boldsymbol{f}$ using ℓ_1 regularization and with $\lambda = 0.3$, $n = 50$, $m = 50$, and $J = 10$.

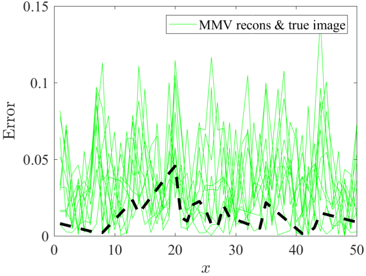

Figure: Error graph illustrating the accuracy of our final reconstruction compared to the accuracy of our MMV measurements.

Using $|||\hat{\boldsymbol{f}} - \boldsymbol{f}|||_1$, we determine that our MMV constructions have an average error of approximately 3.32% from the true image. Similarly, using $|||\hat{\boldsymbol{g}} - \boldsymbol{f}|||_1$, we determine our final reconstruction has an error of approximately 1.32%. Therefore, our final reconstruction is more accurate than our MMV constructions.

Next, we want to test how much multiplicative noise the algorithm can handle. Before, we dampened our Poisson noise by 50%. Now, we test the algorithm with 100% of the Poisson noise.

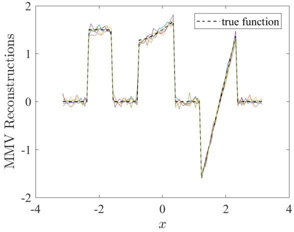

Figure: MMV constructions with $\lambda = 0.3$, $n, m = 100$, and $J = 10$.

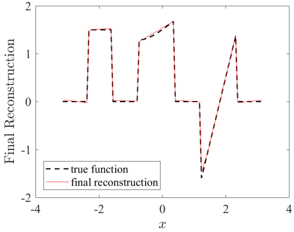

Figure: Final reconstruction with $\lambda = 0.3$, $n, m = 100$, and $J = 10$.

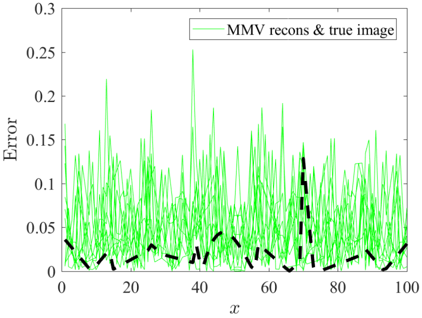

Figure: Error of MMV constructions compared to final reconstruction with 100% of additive Poisson noise.

$$\text{MMV Error} = 4.01\%$$
$$\text{Final Reconstruction Error} = 1.83\%$$

Conclusively, scaling up the Poisson error supplemented to our transform matrix doesn't greatly affect the accuracy of our algorithm. Instead, we can try to manipulate the mean of the Poisson distribution. Before, we were assuming the Poisson error distribution matrix had a **mean of zero**. Now, we see what happens when this matrix instead has a mean of 1. We maintain the Poisson noise at 100%.

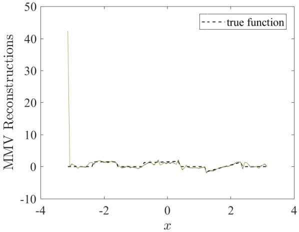

Figure: MMV constructions with $\lambda = 0.3$, $n, m = 100$, and $J = 10$.

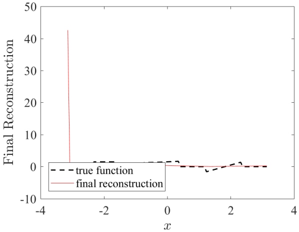

Figure: Final reconstruction with $\lambda = 0.3$, $n, m = 100$, and $J = 10$.

Multiplicative Poisson Noise Reconstruction

Poisson Noise with Mean of 1

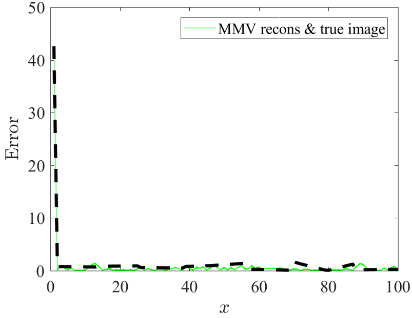

Figure: Error of MMV constructions compared to final reconstruction with our Poisson noise having a mean of 1.

$$\text{MMV Error} = 78.91\%$$
$$\text{Final Reconstruction Error} = 102.38\%$$

Multiplicative Poisson Noise Reconstruction

Poisson Noise with Mean of 0.5

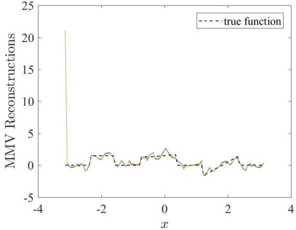

Figure: MMV constructions with $\lambda = 0.3$, $n, m = 100$, and $J = 10$.

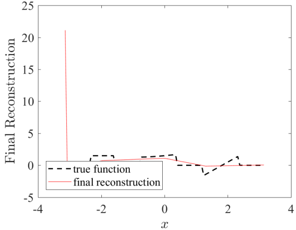

Figure: Final reconstruction with $\lambda = 0.3$, $n, m = 100$, and $J = 10$.

Multiplicative Poisson Noise Reconstruction

Poisson Noise with Mean of 0.5

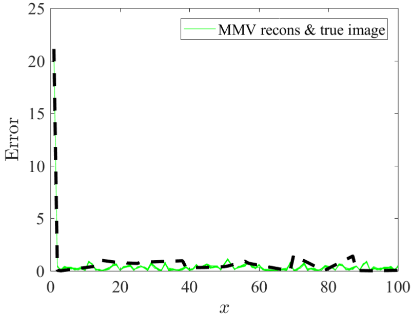

Figure: Error of MMV constructions compared to final reconstruction with our Poisson noise having a mean of 0.5.

$$\text{MMV Error} = 51.85\%$$
$$\text{Final Reconstruction Error} = 71.49\%$$

Multiplicative Poisson Noise Reconstruction

Poisson Noise with Mean of 0.05

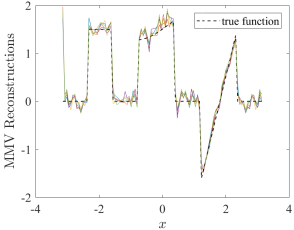

Figure: MMV constructions with $\lambda = 0.3$, $n, m = 100$, and $J = 10$.

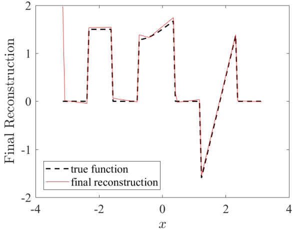

Figure: Final reconstruction with $\lambda = 0.3$, $n, m = 100$, and $J = 10$.

Multiplicative Poisson Noise Reconstruction

Poisson Noise with Mean of 0.05

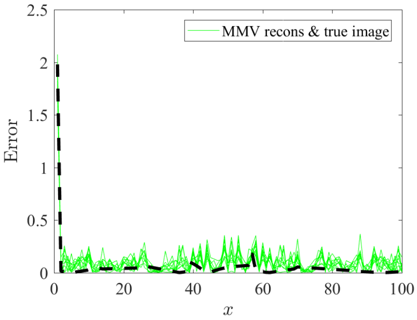

Figure: Error of MMV constructions compared to final reconstruction with our Poisson noise having a mean of 0.05.

$$\text{MMV Error} = 10.12\%$$
$$\text{Final Reconstruction Error} = 5.03\%$$

- This method assumes we have well-sampled data ($N = M$).
- The mean of the multiplicative Poisson noise must be very close to zero.
- As long as the mean of the Poisson noise is exactly equal to zero, the noise will not significantly disrupt the algorithm.
- This method uses the 2^{nd} order polynomial annihilation transform. Higher orders of this method may yield more accurate results.
- Preexisting Gaussian noise $\boldsymbol{\eta}^j$ is assumed to be a random vector. Throughout this algorithm, it has been dampened by 50%.

References

Anne Gelb and Feng Fu. *Mathematics of Misinformation*. Math 76.1. Summer 2018.

Anne Gelb and Theresa Scarnati. "Reducing the Effects of Bad Data Using Variance Based Joint Sparsity Recovery". In: *Journal of Scientific Computing* (2017). DOI: https://doi.org/10.1007/s10915-018-0754-2.

Laura Petto. Personal interviews. June, July, 2018.