

ECE 269 - Final Project

Orthogonal Matching Pursuit

Eric Weise

December 17, 2020

This project is hosted on GitHub: <https://github.com/ericdweise/ece269/blob/master/report.pdf>

1 Background

Orthogonal Matching Pursuit (OMP) is a greedy algorithm that can be used to recover sparse signals. There is a dearth of literature summarizing the usage and limits of OMP for sparse signal recovery. The purpose of this work is to provide and implementation and analysis of OMP. First effectiveness of singal recovery is evaluated in section 3. Then, in sections 4 and 5 the effectiveness of OMP is evaluated on signals with added noise. Finally, in section 6, OMP signal recovery is applied to compressive sampling in images both with and without noise added.

2 The OMP Algorithm

The OMP algorithm is described here. Given a signal $x \in \mathbb{R}^N$ the following steps are taken to recover the signal:

First a random Gausian matrix is created, $A \in \mathbb{R}^{M \times N}$, where $M \ll N$.

The columns of A are normalized.

Calculate $y = Ax$

The first residual is calculated: $r_0 = y$

The support set is initialized as empty: $\Lambda_0 = \{\}$

Main loop:

1. Find the column of A with maximum correlation to residual: $\lambda_k = \arg \max_j |\langle a_j, r \rangle|$
2. Update the support set: $\Lambda_k = \Lambda_{k-1} \cup \lambda_k$
3. Find the recovered signal \hat{x}_k , which is the Least Squares solution to $A_{\Lambda_k} \hat{x}_k = y$
4. Update the residual: $r_{k+1} = y - \Lambda_k \hat{x}_k$

This loop will continue until one of two conditions is met:

1. Given a threshold we can stop when $\|r\|_2$ is less than this threshold.
2. Provided that the sparsity level, s , of x is known, we can stop after the loop executes s times.

The recovered signal is \hat{x}_k , where k is the final iteration count of the main loop.

Error Analysis

Given a signal x and its recovered analog \hat{x} , the normalized error is calculated as:

$$\frac{\|x - \hat{x}\|_2}{\|x\|_2}$$

Implementation

The implementation of the OMP used in this project can be found here:

<https://github.com/ericdweise/ece269/blob/4e869d18719761ae3458ef0244da7af4768095fc/tools.py#L148>

3 Noiseless Signal Recovery (*part c*)

The Exact Signal Recovery (ESR) rate is very highly correlated to s , and somewhat correlated to M . This is evident in the sharp increase in the Probability of ESR for low s . For all matrix sizes the probability of ESR is zero for a sparsity level above $\frac{N}{10}$, but this threshold decreases slightly as the height of the matrix, M decreases.

The Normalized Error plots show roughly the inverse trend as that in the ESR plots: The Error rate is low for small sparsity, and reach a plateau close to 1.2.

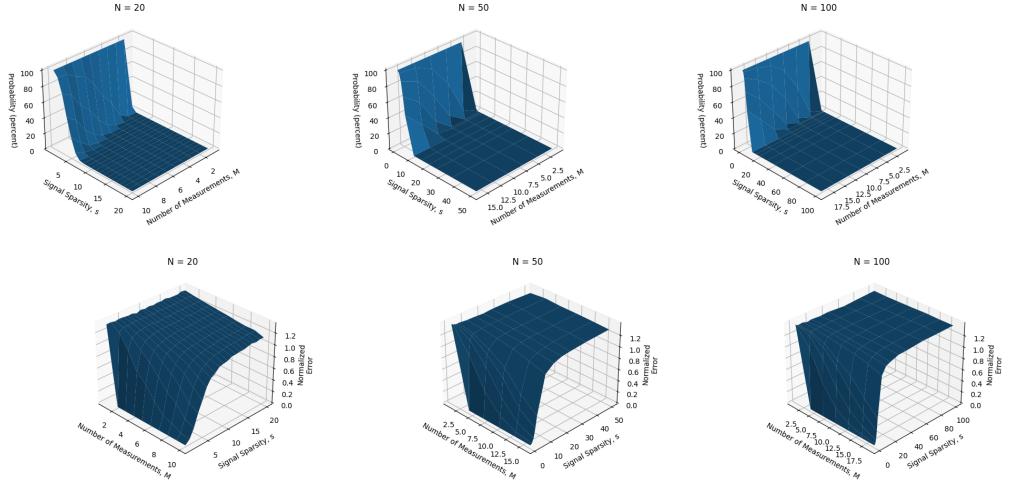


Figure 1: Noiseless support set recovery. Top row: The probability of Exact Support Recovery. Bottom row: Normalized error of recovered signal. NOTE: The ESR and the error plots are viewed from different angles.

4 Noisy Signal Recovery With Known Sparsity (Part d-i)

In this experiment the same procedure was applied in the noiseless signal recovery, but two major differences were made:

1. After the original sparse signal x is generated, random noise is added to every element in x . This resulting noisy vector, x_{noise} is used to calculate $y = Ax_{noise}$
2. The OMP loop is terminated after s iterations, where s is the sparsity level of x .

Two experiments are performed. The first experiment adds noise drawn from a normal distribution with mean 0 and variance 0.05. The second experiment adds noise drawn from a normal distribution with mean 0 and variance 1.

It should be noted that the Normalized error was calculated using the original signal, not the noisy signal.

As we might predict, adding small amounts of noise changes the ESR and Normalized Error plots slightly, while adding more noise significantly degrades the probability of ESR. For a noise variance of 0.05 the ESR and Normalized error plots are almost identical to the noiseless case. However, for a noise variance of 1 the probability of ESR decreases significantly, and never rises above 75%. The normalized errors in the variance 1 case are also significantly higher, never dropping lower than 0.5, and plateauing at 1.4, an increase of about 15% over noiseless recovery.

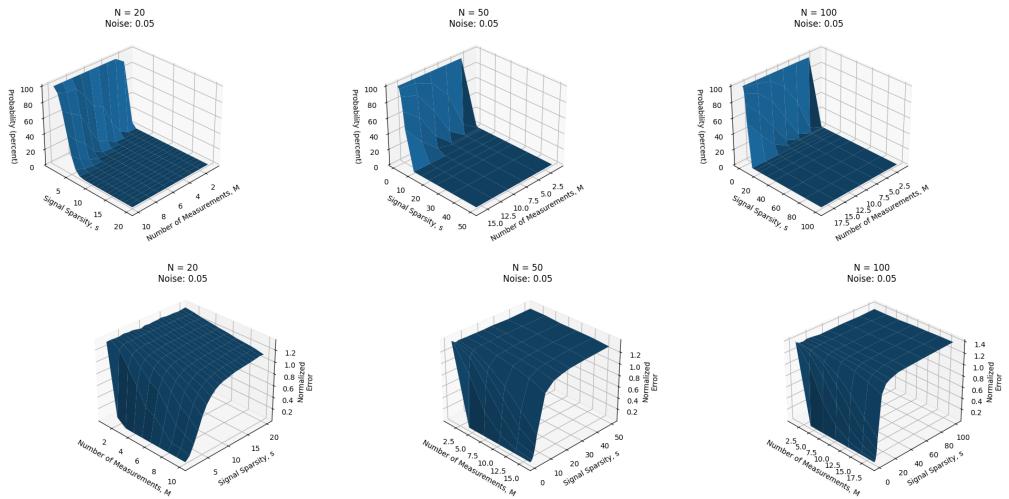


Figure 2: Noisy support set recovery **variance=0.05**. Stop condition achieved when loop count is the sparsity of the input signal. Top row: The probability of Exact Support Recovery. Bottom row: Normalized error of recovered signal. NOTE: The ESR and the error plots are viewed from different angles.

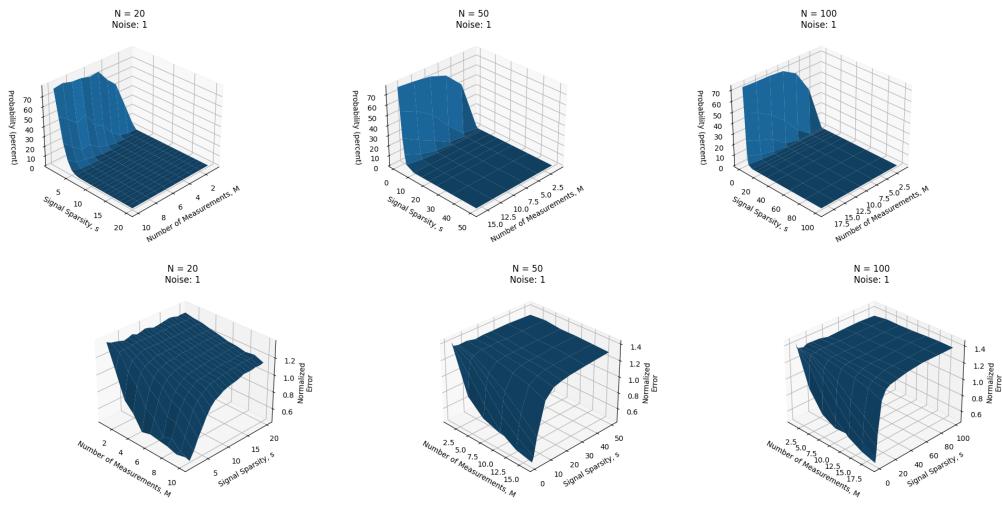


Figure 3: Noisy support set recovery **variance=1**. Stop condition achieved when loop count is the sparsity of the input signal. Top row: The probability of Exact Support Recovery. Bottom row: Normalized error of recovered signal. NOTE: The ESR and the error plots are viewed from different angles.

5 Noisy Signal Recovery (part d-ii)

In this experiment the same procedure was applied in the previous section, except that the stop condition is changed:

1. The OMP loop is terminated when the two norm of the residual, $\|r_k\|_2$ is less than 0.001.

Two experiments are performed. The first experiment adds noise drawn from a normal distribution with mean 0 and variance 0.05. The second experiment adds noise drawn from a normal distribution with mean 0 and variance 1.

It should be noted that the Normalized error was calculated using the original signal, not the noisy signal.

As might be expected, the results are similar identical to the previous section. This stands to reason because if exact support recovery does not happen after s iteration of the main loop, it will not happen after more iterations, That is: the agreement between x and \hat{x} will not increase if the loop count exceeds the sparsity level of s .

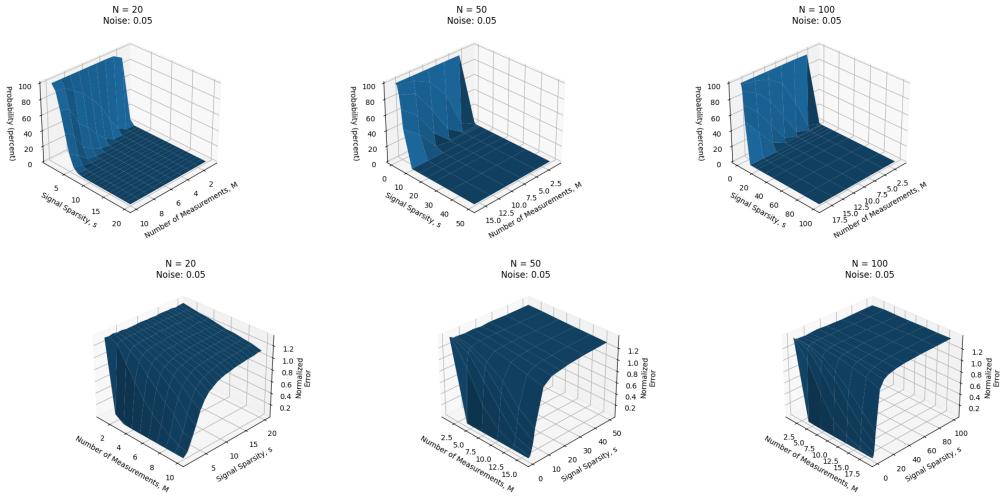


Figure 4: Noisy support set recovery **variance=0.05**. Stop condition achieved when the 2-norm of residual is less than 0.001. Top row: The probability of Exact Support Recovery. Bottom row: Normalized error of recovered signal. NOTE: The ESR and the error plots are viewed from different angles.

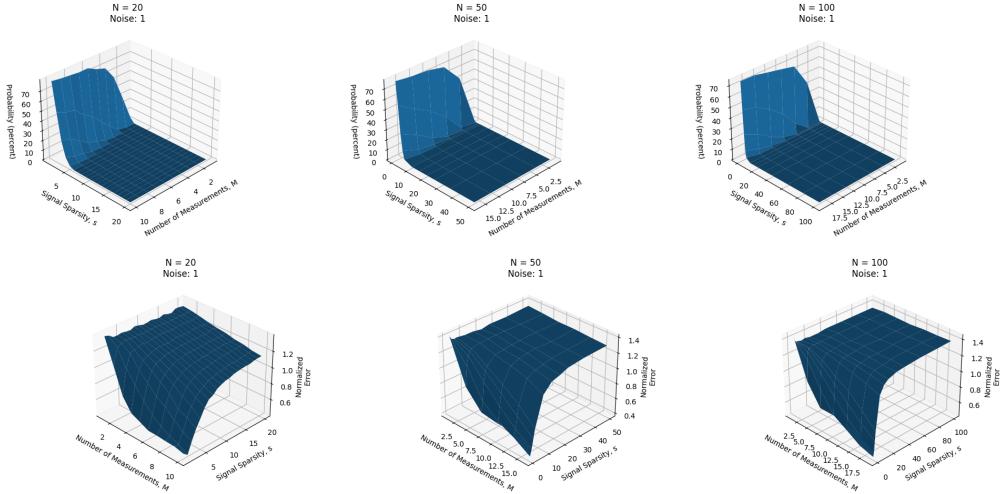


Figure 5: Noisy support set recovery **variance=1**. Stop condition achieved when the 2-norm of residual is less than 0.001. Top row: The probability of Exact Support Recovery. Bottom row: Normalized error of recovered signal. NOTE: The ESR and the error plots are viewed from different angles.

6 Image Recovery (Part d-iii)

Compressive Sampling and Signal Recovery Methods

Compressive sensing is a rich field. The goal is to take the fewest number of samples and maintain recoverability of the source signal. First we must find an appropriate orthonormal basis ψ_1, \dots, ψ_N . A signal x can be created as a linear combination of the elements ψ_i . The sparse sensing happens in this basis, choosing the highest correlated components in this orthonormal basis to the original signal. Recovery of the sparse signal, x_s , the sparse representation of x in the Ψ basis, can be performed using OMP. In this paradigm, Ψ is the matrix whose columns are the O.N. basis, ψ_1, \dots, ψ_N ,

The purpose of this experiment is to demonstrate the recoverability of a compressed image. The image will be compressed using and recovered for different sparsity levels.

These are the steps used in Compressive Sensing and Signal Reconstruction:

1. First, a random gaussian matrix matrix is made, $\Phi \in \mathbb{R}^{M \times 64}$.
2. The image is broken into 8×8 subimages.
3. The subimage is transformed into frequency space using a Discrete Cosine Transform. The 64×64 DCT matrix forms Ψ .
4. The transformed 8×8 subimage is rearranged into a vector $x \in \mathbb{R}^{64}$. The order uses the zig-zag pattern demonstrated in *figure 6* to bias low frequency coefficients in Fourier space.
5. **Compressive Sensing** The largest s elements in x are retained. The rest are set to zero. This sparse vector is denoted x_s .
6. Calculate $y = \Phi x_s$. ($y \in \mathbb{R}^M$)
7. Perform OMP to recover $\hat{x} \in \mathbb{R}^{64}$. The stop condition is set to $\|\Phi \hat{x} - y\|_2 < 10^{-3}$
8. \hat{x} is rearranged to an 8×8 array using the zig-zag ordering.
9. The Inverse Discrete Cosine Transform, Ψ^{-1} , is applied to convert the recovered subimage back to image space.

In the above steps $y = \Phi \Psi x_s$ is the first residual used in OMP to reconstruct the signal, x . This process is repeated for every 8×8 subimage. For this experiment it was found that a Gaussian random matrix Φ with size 30×64 was both time performant and recovered images with good detail.

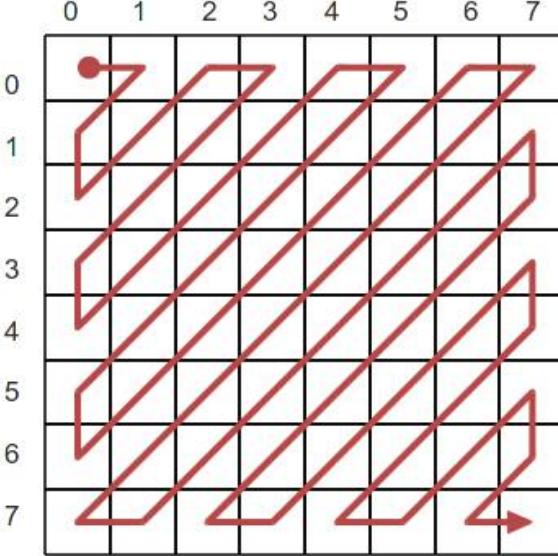


Figure 6: The Zig-zag ordering. In the DCT Basis this preferences low frequency components over higher frequency components.

Error Measurements

To measure the efficacy of the recovered image the Peak Signal to Noise Ratio (PSNR) is calculated. PSNR is used on the entire, large image and its reconstructed analog, not to the 8×8 subimages. The PSNR is calculated as:

$$PSNR = 10 \cdot \log_{10} \frac{MAX^2}{MSE} \quad (1)$$

where

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i, j) - K(i, j)]^2 \quad (2)$$

is the mean squared error of the images, I and K are the original and recovered images, m, n are the number of rows and columns in the images, and MAX is the maximum possible value a pixel can have.

Image Recovery After Compressive Sensing

This experiment was performed on four images. The images were chosen to show a varying level of complexity. The recovered images and some recovered images are shown in figures 7-10.

Even at a very low sparsity level there is significant image reconstruction. At $s/N = 2/64$ we begin to see some texture, although finer details are not recovered. However, even at low sparsity levels, around 8, we can see more details being recovered. Notice the texture in the elephant's skin, the tree branches under the arch, and the hairs of the koala start to appear.

In figure 11 we can see the PSNR plotted as a function of sparsity in the measurement basis for each of the images. In all but one of the images, the PSNR increases as the measurements become less sparse. The complex images, elephant, koala, and arch, have lots of detail, which correspond to high frequency components in the DCT space. However, the recovery tends to plateau between a sparsity level of 4 and 8. This corresponds to a compression level between 6.25% and 12.5%.

Interestingly the spiral has peaks at a sparsity level of 3. This is because the image has large regions where the pixels change very slowly in real space. This means that they can be represented using very few vectors in the DCT basis. Approximation of these regions with higher frequency vectors will degrade the image in real space.



Figure 7: The original image of the arch (left) recovered image with $s=2$ (center) and recovered image with $s=8$ (right)



Figure 8: The original image of elephant (left) recovered image with $s=2$ (center) and recovered image with $s=8$ (right)



Figure 9: The original image of the koala (left) recovered image with $s=2$ (center) and recovered image with $s=8$ (right)



Figure 10: The original image of the spiral (left) recovered image with $s=2$ (center) and recovered image with $s=8$ (right)

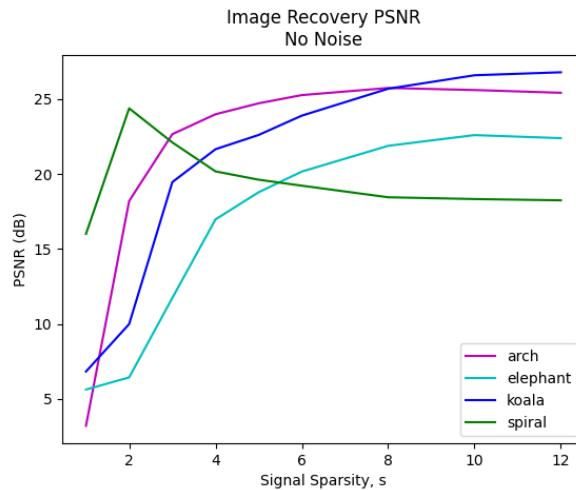


Figure 11: PSNR measured for varying sparsity levels and for different levels of added noise.

Image Recovery With Added Noise

The process of reconstructing the noisy image is the same as the previous section. Before the process of compressive sensing begins, however, random noise is added to the image. Each pixel has added a random number drawn from a normal distribution. This is done twice, first with a variance of 10, then again with a variance of 50. The mean for both experiments is zero. Since the images have a bit depth of 2^8 , a variance of 10 has a noise level of approximately 2.5%, and a variance of 50 has a noise level of approximately 25%.

The error measurements are also the same as noiseless image recovery. It should be noted that the PSNR is calculated using the clean original image. That is, the image with no added noise.

Figures 12 and 13 show the result of reconstructing the noisy koala image for the two noise levels. It is evident, especially in figure 13, that the recovered image is less noisy for more sparse measurements. When using a sparsity level of 2 in the measurement basis stores more information from the original image than a sparsity level of 8. This is again evident in figure 14, which plots the PSNR vs sparsity level for the different noise levels. This is because the recovery biases lower frequency coefficients in the DCT domain, which selects out the high frequency Gaussian noise. However, at a certain sparsity level the PSNR starts to decrease as the reconstructed image starts to include the high frequency noise.



Figure 12: Results for image reconstruction with added noise **variance=10**. From left to right: Image with added noise. Recovery with sparsity 2. Recovery with sparsity 8.



Figure 13: Results for image reconstruction with added noise **variance=50**. From left to right: Image with added noise. Recovery with sparsity 2. Recovery with sparsity 8.

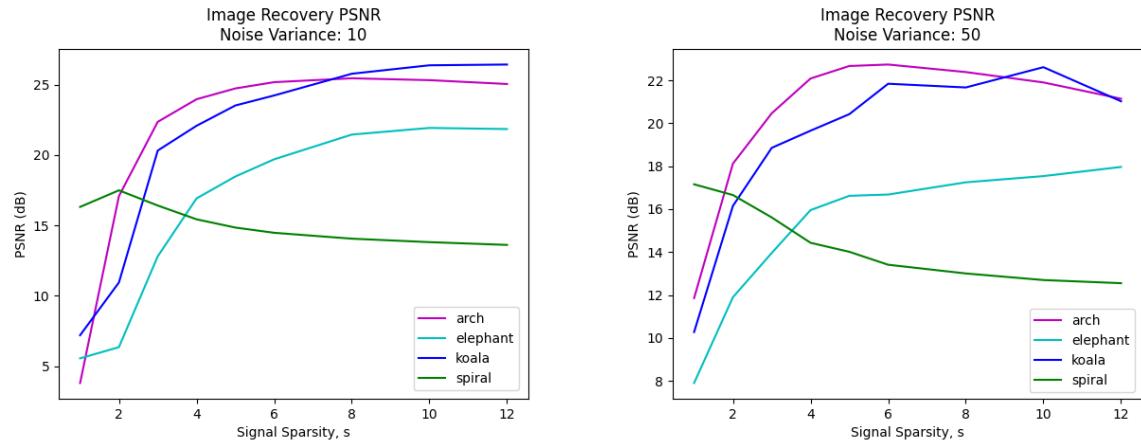


Figure 14: Peak Signal to Noise Ratio as a function of sparsity for noisy images. Left noise variance=10, right noise variance=50.