OPTIMIZING SPARSE IMAGE RECONSTRUCTION

Jianhang Liu, Gore Kao, Yi-Jing Sie
Carnegie Mellon University
Pittsburgh, PA, USA
{jianhanl,gorek,ysie}@andrew.cmu.edu

1 INTRODUCTION

In imaging problems, an original image of interest can be corrupted by noisy measurements, leading to a deteriorated observed image. The goal of image reconstruction is to then recover the original signal as closely as possible. These imaging problems can often be formulated as an inverse problem involving the recovery of an image $x^* \in \mathbb{R}^n$ from n noisy measurements. This can be expressed as an inverse problem $y = Ax^* + e$ where $y \in \mathbb{R}^m$ is the observed corrupted image, $A \in \mathbb{R}^{mxn}$ is the measurement operator and $e \in \mathbb{R}^m$ is possible added noise.

In this project, we explore sparse image reconstruction where we assume the observed image is sparse under some domain and we seek to recover this sparse vector. Specifically, given an image corrupted by an unknown mask creating random regions of blank pixels, we aim to recover the original image utilizing optimization techniques (Section 3 below). This task can be compared to compressed sensing techniques which aim to recover a sparse vector x^* from $m < n$ measurements. We assume some prior properties of the measurement mask and compare a few algorithms to observe the image reconstruction quality.

2 APPROACH

2.1 WAVELET TRANSFORM

By using the wavelet transform, we can convert the 2-D image data matrix into a 1-D K-sparse data vector when picking the top K wavelet coefficient from results, which will then be fed into various image recovery algorithm described in later sections. The core principle of wavelet transform is do iterative splitting on data into two parts by applying high-pass filters (HPF) : h and g. The h part is the part that has been filtered out by HPF, while the g part is the part that will be kept after applying HPF. For starting next iteration, we will pick the g part and apply a new HPF which will have higher cut-off frequency, and do the iteration multiple times. The result will have only one g part and multiple h parts. At this time, we will pick up top K coefficients among the h part for doing the following optimization and recovery work.

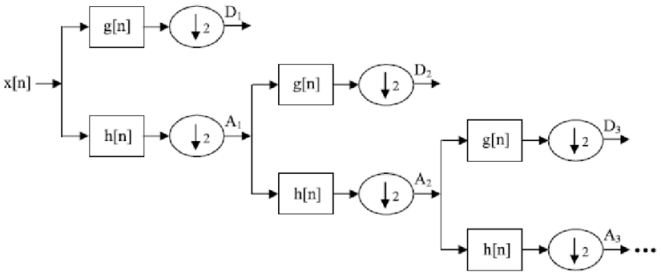


Figure 1: Wavelet Transform Structure

2.2 SPARSE OPTIMIZATION

In this project, we recovered the original image by solving the sparse optimization problem, which is defined as follows:

$$\min_{x \in \mathbb{R}^n} f(x)$$
$$\text{s.t. } ||x||_0 \leq s$$

Here $f : \mathbb{R}^d \to \mathbb{R}$ is assumed to be differentiable and is the loss function that we would like to minimize. The sparsity constraint $||x||_0 \leq s$ requires that at most s many number entries of the solution vector x are non zero. We adopted iterative thresholding approach where an iterative step alternates between gradient steps and thresholding steps:

$$\text{An iterative step : } x_{t+1} = \Psi_s(x_t - \alpha_t \nabla f(x_t))$$
$$\text{where } \begin{cases} \text{Gradient step: } x'_{t+1} = x_t - \alpha_t \nabla f(x_t) \\ \text{thresholding step: } x_t + 1 = \Psi_s(x'_{t+1}) \end{cases}$$

In the grading step, we lower the value of the loss function; in the thresholding step, we use thresholding operator $\Psi_s(x)$ to enforce sparsity constraints on x.

3 METHODS

We consider three methods for optimizing sparse image reconstruction: Soft Thresholding, Hard Thresholding, and Reciprocal Thresholding.

3.1 SOFT THRESHOLDING

The first optimization methods we used here to recover corrupted image is (Iterative) Soft Thresholding, which will be under sparse Wavelet transform domain. After performing the wavelet transform, the image data will become sparse, and we pick the most significant wavelet coefficients while setting the rest to 0. After that, we performed Iterative Soft Thresholding on those data points. We iterate for 1000 step or until the small residue

$$\frac{\|y - A(x^{(k)})\|}{\|y\|}$$

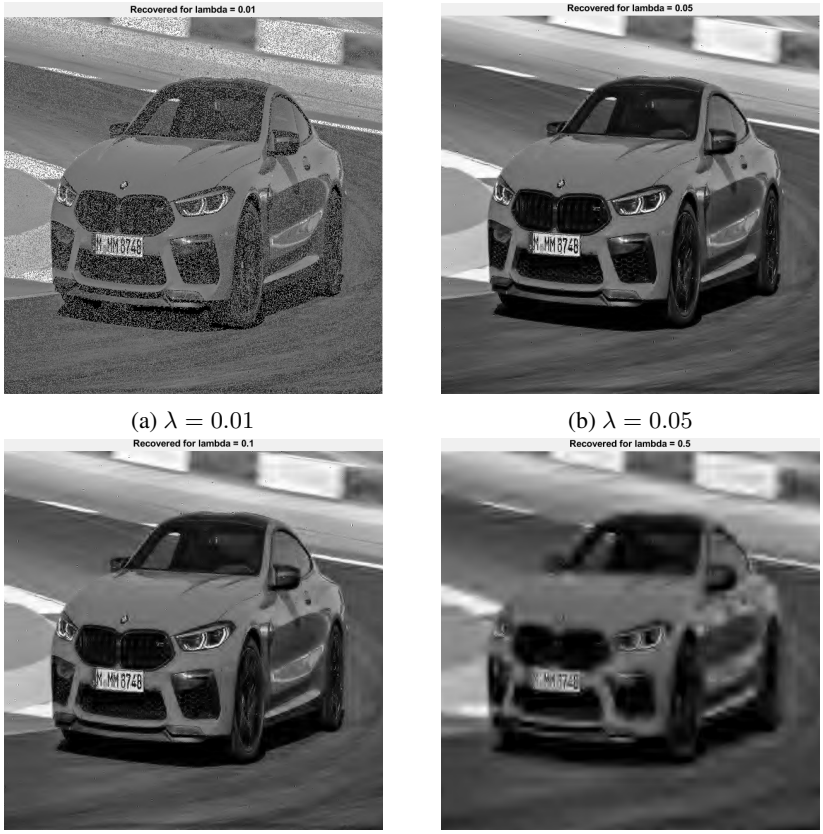
between the measurement operator applied on the recovered image and corrupted image is smaller than the tolerance (here we are using tolerance = 0.1). The Soft Thresholding operator here is defined as:

$$(\Psi_s^{ST}(z))_i = \begin{cases} z_i - \lambda, & z_i > \lambda \\ 0, & |z_i| \leq \lambda \\ z_i + \lambda, & z_i < -\lambda \end{cases} \tag{1}$$

But instead of adjust the input z by adding or subtracting λ only, we will multiply λ with an extra parameter $\eta = 0.1$ here to reduce the step size. It is expected that the value of λ will affect final result of the IST algorithm. We compare the reconstructed images under varying parameter $\lambda = 0.01, 0.05, 0.1, 0.5$ (Figure 2).

From the recovered images in Figure 2 with different λ, we conclude that when λ is between $0.05 \sim 0.1$, the algorithm has best performance. Another detail to note is that IST will have better recovery performance on lower-frequency region on the original image (e.g., windsheld in example) while having worse performance on high-frequency region (e.g, car wheel in the example). Therefore, when we subtract the recovered image from the original image, the remainder will be the high frequency part of image, which is a pretty efficient edge detection method.

In order to use IST method to recover image, we have to know the mask, or guess the mask if we are not given. In the case that mask is not given, we designed our mask guessing algorithm. The principle is that, we will first compare



(a) $\lambda = 0.01$ (b) $\lambda = 0.05$

(c) $\lambda = 0.1$ (d) $\lambda = 0.5$

Figure 2: Reconstructed Images using IST for different Lambdas

random two masked images, and figure out the black (mask) pixels that exists in both image. Since the mask is generated randomly in our case, it is relatively less likely to exist a large continuous mask matrix (i.e., mask is less likely to have 3x3, 4x4 black matrix). Therefore, we did the sparse analysis for all potential mask pixels obtained above, for each potential mask pixels, if the surrounding 8 pixels (3x3 here) are all black pixels. If yes, then we assigned that pixel as "true mask pixel", otherwise, it will be classified as normal black pixel from original image.

3.2 Hard Thresholding

With a given corrupted image that is the result of a measurement mask applied to an original image of interest, we can write the observation formulation as $y = Ax$, where y is observed corrupted image, A is the measurement operator, and x is the original image.

For Hard Thresholding (HT) Blumensath & Davies (2009); Jain et al. (2014), we constrain the optimization to have an l_0 sparse constraint. We assume the image is sparse in the wavelet domain and define a variable, K, such that the image in the wavelet domain is constrained to be K-sparse. This constrains the wavelet image coefficients to have at most K non-zero elements. Letting Ψ/Ψ^* denote the wavelet and inverse wavelet operators respectively, we can therefore formulate this as a minimization problem to optimize:

$$\min_x \|y - Ax\|^2 \text{ s.t. } \|\Psi x\|_0 \leq K$$

To simplify the notation of the constraint slightly, we can let $S = \Psi x$ which results in the following:

$$\min_S \|y - A\Psi^* S\|^2 \text{ s.t. } \|S\|_0 \leq K$$

Finally for simplification, let $B = A\Psi^*$ to get the optimization problem:

$$\min_S \|y - BS\|^2 \text{ s.t. } \|S\|_0 \leq K$$

For the HT implementation, we keep the top K largest magnitude coefficients while setting the remaining coefficients to 0. The formal Hard Thresholding (HT) operator is therefore defined as

$$(\Psi_s^{HT}(z))_i = \begin{cases} z_i & i \in S \\ 0 & i \notin S \end{cases} \tag{2}$$

Where S is the support set containing indices of the top K largest magnitude entries of z. From the defined HT operator, we can apply this operator on top of a gradient step to get the next iterative step. Therefore applying Equation 2 for each iteration, the Iterative Hard Thresholding (IHT) operation is defined as:

$$x_t = \Psi_s^{HT}(x_{t-1} - \eta \nabla f(x_{t-1}))$$

To overview our implementation of reconstruction using Iterative Hard Thresholding (IHT), we use the wavelet transform to make images sparse, applying IHT on the wavelet coefficients, and then use the inverse wavelet transform to get back the reconstructed image. We run the procedure for 1000 iterations, keeping top 10% of coefficients. A sample reconstruction using IHT is shown in Figure 3. Further reconstruction examples can be found in Section 5.

(a) Corrupted Image (b) Reconstructed Image

Figure 3: Corrupted and Reconstructed Images using IHT

3.3 Reciprocal Thresholding

After seeing the results from Soft Thresholding and Hard Thresholding, one may wonder about other thresholding operators lying between these two. To find a suitable thresholding algorithm for our image reconstruction problems, we studied the paper published by Liu & Foygel Barber (2020). The authors note that due to the nonconvexity of the sparsity constraint $||x||_0 \leq s$, iterative thresholding operaters cannot guarantee to find the global minimum without strong assumptions on $f(x)$. Therefore, the paper proposed a notion of restricted optimality for assessing iterative thresholding methods: Given a iterative thresholding algoritm that keeps the sparsity of the input x at s for f(x), we say it achieves restricted optimality guarantee if it satisfies the following inequality:

$$\lim_{t \to \infty} f(x(t)) \leq \min_{||x||_0 \leq s'} f(x)$$

for some tighter sparsity constraint $s' \leq s$

In other words, restricted optimality guarantees that an s-sparse iterative thresholding algorithm performs well relative to a more restrictive s' sparsity constraint, and the paper assessed an iterative thresholding operator based upon its ability to guarantee restricted optimality.

Furthermore, the paper developed the notion of relative concavity of a thresholding operator that can fully categorize any thresholding operator on its restricted optimality guarantee:

Let $s \in \{1, ..., d\}$ be any fixed sparsity level and define the sparsity proportion $\rho = \frac{s'}{s} \in (0, 1]$. The relative concavity of an s-sparse thresholding operator Ψ_s relative to sparsity proportion ρ is

$$\gamma_{s,\rho}(\Psi_s) = \sup\Big\{ \frac{\langle y - \Psi_s(z), z - \Psi_s(z)\rangle}{||y - \Psi_s(z)||^2} \;\; y, z \in \mathbb{R}^d,\, ||y||_0 \leq \rho s,\, y \neq \Psi_s(z) \Big\} \tag{3}$$

Also, the lower bound for all thresholding operators:

For any map $\Psi_s : \mathbb{R}^d \to \{x \in \mathbb{R}^d : ||x||_0 \leq s\}$ and any sparsity $\rho \in (0, 1]$,the relative concavity is lower-bounded as

$$\gamma_{s,\rho}(\Psi_s) \geq \frac{\rho}{1 + \rho}$$

Motivated by the optimal value of relative concavity, the paper proposed a thresholding operator with a parameter c whose relative concavity matches this lower bound with a proper choice of c :

Given a vector $z \in \mathbb{R}^d$, let $S \subset \{1, ..., d\}$ be the indices of the largest magnitude s entries of z and $\tau = \max_{i \notin S} |z_i|$ be the magnitude of the $(s + 1)$-st largest entry of z. Then the reciprocal thresholding $\Psi_s^{RT,c}(z)$ operates entry-wise as follows:

$$\Big(\Psi_s^{RT,c}(z)\Big)_i = \begin{cases} \text{sign}(z_i) \cdot (\frac{1}{2}|z_i| + \frac{1}{2}\sqrt{|z_i|^2 - \tau^2(1 - c^2)}, & \text{if } i \notin S \\ 0, & \text{if } i \in S \end{cases} \tag{4}$$

$\Psi_s^{RT,c}$ with $c = \rho$ has the relative concavity $\frac{\rho}{1+\rho}$, which is exactly the optimal value of relative concavity. Taking $c = 1$ yields $\Psi_s^{RT,1} = \Psi_s^{HT}$, the hard thresholding operator, and taking $c = 0$ defines the "universal" reciprocal thresholding operator $\Psi_s^{RT,0} = \Psi_s^{RT}$ in the sense that it performs nearly well across all sparse thresholding problems:

For any $z \in \mathbb{R}^d$, Ψ_s^{RT} operates element-wise as

$$\Big(\Psi_s^{RT,c}(z)\Big)_i = \begin{cases} \text{sign}(z_i) \cdot (\frac{1}{2}|z_i| + \frac{1}{2}\sqrt{|z_i|^2 - \tau^2}), & \text{if } i \notin S \\ 0, & \text{if } i \in S \end{cases} \tag{5}$$

For the purpose of our project, we used "universal" reciprocal thresholding to recover the original image with the sparsity constraint be 10% of the size of the original image. More specifically, we first employed the wavelet transform to sparsify the original image, apply "universal" reciprocal thresholding on the wavelet coefficients, and then used the inverse wavelet transform to derive the reconstructed image. The entire process is run for 1000 iteration.

A demonstration of an image reconstruction result using "universal" reciprocal thresholding is shown in Figure 4. Further reconstruction examples can be found in Section 5.

(a) Corrupted Image (b) Reconstructed Image

Figure 4: Corrupted and Reconstructed Images using "universal" reciprocal thresholding

4 General Algorithm

From the defined thresholding operator methods defined above in Section 3, we formulate a general algorithmic procedure to follow for image reconstruction given a thresholding operator, T.

Algorithm 1: Iterative Thresholding

$x_0 \leftarrow$ random initialization;
$k \leftarrow 0$;
while $k <$ *number of iterations* **do**
  $k \leftarrow k + 1$;
  $x_k \leftarrow x_{k-1} - \eta A^*(Ax_k - y)$;
  wavelet_x$_k \leftarrow$ wavelet$\{x_k\}$;
  wavelet_x$_k \leftarrow T\{$wavelet_x$_k, \Theta\}$;
  $x_k \leftarrow$ iwavelet{wavelet_x$_k$};
end
return x_k;

In Algorithm 1, η is the step size, A/A^* are the measurement/adjoint measurement operators respectively, and Θ is any additional parameters used for the thresholding operator. A uniform random distribution is used for the initialization step in the range of the pixel values. For our implementation, we perform 1000 iterations for each thresholding method.

5 Results

In this section we report our qualitative and quantitative results for five example images.

5.1 Image Reconstructions

For each image we show the original, corrupted, and reconstructed images for each method described in Section 3. The results shown utilize $\lambda = 0.1$ for IST and $c = 0$ for Reciprocal Thresholding. The process is run for 1000 iterations.

(a) Original

(b) Corrupted

(c) Soft Thresholding

(d) Hard Thresholding

(e) Reciprocal Thresholding

Figure 5: Reconstruction Results for Image 1

(a) Original

(b) Corrupted

(c) Soft Thresholding

(d) Hard Thresholding

(e) Reciprocal Thresholding

Figure 6: Reconstruction Results for Image 2

(a) Original

(b) Corrupted

(c) Soft Thresholding

(d) Hard Thresholding

(e) Reciprocal Thresholding

Figure 7: Reconstruction Results for Image 3

(a) Original

(b) Corrupted

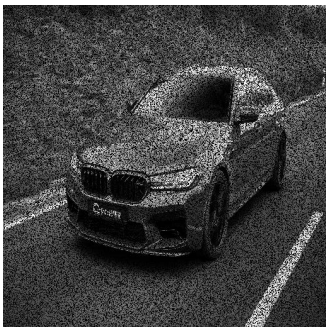
(c) Soft Thresholding

(d) Hard Thresholding

(e) Reciprocal Thresholding

Figure 8: Reconstruction Results for Image 4

(a) Original



(b) Corrupted

(c) Soft Thresholding

(d) Hard Thresholding

(e) Reciprocal Thresholding

Figure 9: Reconstruction Results for Image 5

5.2 Mean Squared Error Reconstructions

We utilize the Mean Squared Error (MSE) metric to compare the error between original and reconstructed images. This measures the average squared difference between the estimated and actual images and is defined as follows.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \tag{6}$$

n is the number of pixels, Y_i is the ground truth pixel value from the original image, and $\hat{Y}_i$ is the predicted reconstructed pixel value. For consistency, we normalize all values to be between 0 and 1 before calculating MSE.

	Image 1 (Fig 5)	**Image 2** (Fig 6)	**Image 3** (Fig 7)	**Image 4** (Fig 8)	**Image 5** (Fig 9)
Soft Thresholding	0.0303	0.0219	0.0242	0.0187	0.0083
Hard Thresholding	0.0483	0.0221	0.0379	0.0238	0.0152
Reciprocal Thresholding	0.0250	0.0071	0.0135	0.00856	0.0074

Table 1: MSE Reconstruction Errors

References

Thomas Blumensath and Mike E Davies. Iterative hard thresholding for compressed sensing. *Applied and computational harmonic analysis*, 27(3):265–274, 2009.

Prateek Jain, Ambuj Tewari, and Purushottam Kar. On iterative hard thresholding methods for high-dimensional m-estimation. *Advances in neural information processing systems*, 27, 2014.

Haoyang Liu and Rina Foygel Barber. Between hard and soft thresholding: optimal iterative thresholding algorithms. *Information and Inference: A Journal of the IMA*, 9(4):899–933, 2020.