ECE566 Computational Inference Project Final Report

Ufuk Soylu, Berk Iskender

1 Abstract

In this project final report, we study five different algorithms ISTA, FISTA, ADMM, PnP and LISTA for solving linear inverse problems arising in signal processing - especially in image processing and imaging - in LASSO setting with the theoretical aspects of the problem itself. Firstly, we provide essential information related to the LASSO and its relation with "ideal" sparse regularization and problem settings that we work with. Then, we analyze the algorithms both theoretically and empirically by providing their theoretical convergence proofs, theoretical convergence rates, lower bounds for respective rates and empirical convergence rates to verify the lower bounds for convergence. In the final part, we conduct experimental verification of the results we find and implement these algorithms for two different problem settings: (i) sparse recovery setting and (ii) image deblurring setting to obtain a fair comparison. In the first setting, the true signal is sparse and we conduct experiments with different noise levels to recover the true signal from a measurement signal which is obtained using a random Gaussian matrix. In the second setting, the true signal is sparse in a different domain (in its Wavelet transform) and we experiment with different noise levels to recover the true signal from a measurement signal which is obtained using a Gaussian blur kernel and a random Gaussian matrix respectively. Finally, we compare these algorithms under these settings to reveal their strengths and weaknesses.

Contents

1 Abstract 1

2 Introduction 2
2.1 Bayesian Interpretation of LASSO 2
2.2 Relation with "ideal" sparse regularization 2
2.3 Unique recovery of x and measurement bound on A 3
2.4 LASSO with wavelet transform 4

3 Iterative Shrinkage Thresholding Algorithm (ISTA) 4
3.1 Algorithm Analysis 4
3.2 Theoretical Convergence Analysis 5

4 A Fast Iterative Shrinkage Thresholding Algorithm (FISTA) 6
4.1 Algorithm Analysis 6
4.2 Theoretical Convergence Analysis 6

5 Learned ISTA (LISTA) 7
5.1 Algorithm Analysis 7
5.2 Theoretical Convergence Analysis 8

6 Alternating Direction Method of Multipliers (ADMM) 9
6.1 Algorithm Analysis 9
6.1.1 Scaled Form of ADMM 9
6.1.2 ADMM application for Generalized LASSO 10
6.1.3 Theoretical Convergence Rate Analysis 10

7 Plug and Play (PnP) Priors for Model Based Reconstruction 11

7.1 Algorithm Analysis 11

7.1.1 MAP cost function for solving inverse problems 12

7.2 Theoretical Convergence Analysis 12

8 Computational Costs of Algorithms 13

9 Computational Experiments 14

9.1 Sparse recovery: Measurement Bound Experiments with LASSO 14

9.1.1 Robust CS problem - Noiseless measurement - Random Gaussian Measurement Matrix RIP condition analysis 14

9.2 Empirical Analysis of the Theoretical Convergence Rate/Analysis for provided algorithms for LASSO problem 14

9.2.1 Selection of λ 14

9.2.2 ISTA 15

9.2.3 FISTA 16

9.2.4 LISTA 16

9.2.5 ADMM 17

9.2.6 PnP 18

9.3 Empirical comparison of algorithms on the different problem settings 19

9.3.1 Problem 1: Sparse recovery 19

9.3.2 Problem 2: LASSO with wavelet decomposition prior 21

10 Discussion 22

11 Conclusion 24

2 Introduction

Least Absolute Shrinkage and Selection Operator (LASSO), is a regression analysis method that performs variable selection and regularization simultaneously to enhance the prediction accuracy and interpretability of the model it produces. It solves a non-smooth convex optimization problem.

Problem attempts to find a vector $x \in \mathbb{R}^n$, given an observation (measurement) vector $b \in \mathbb{R}^m$, a measurement matrix $A \in \mathbb{R}^{m \times n}$ and a constant $\lambda > 0$ that minimizes

$$F(x) \triangleq \frac{1}{2}\|b - \mathrm{A}x\|^2 + \lambda\|x\|_1 \tag{1}$$

The first term in (1) represents the data fidelity and the second term in (1) is the $l1$-norm of x.

2.1 Bayesian Interpretation of LASSO

The optimization problem that we are solving in LASSO can be interpreted using Bayesian approach. If we take $p(b|x) \sim \frac{1}{(2\pi\sigma^2)^n} exp(-\frac{(b-Ax)^T(b-Ax)}{2\sigma^2})$ and $p(x) \sim \frac{1}{2}exp(-|x|)$, minimizing $F(x)$ is equivalent to the following MAP problem:

$$\begin{aligned} \max_x \ln p(x|b) = \max_x \ln p(b|x) + \ln p(x) &= \max_x -\frac{n}{2}\ln\left(2\pi\sigma^2\right) - \frac{1}{2\sigma^2}\|b - \mathrm{A}x\|^2 - n\ln 2 - \|x\|_1 \\ &= \min_x \frac{1}{2\sigma^2}\|b - \mathrm{A}x\|^2 + \|x\|_1 \end{aligned} \tag{2}$$

where $\lambda = 2\sigma^2$ in (1).

2.2 Relation with "ideal" sparse regularization

In the project, we will solve LASSO problem with the respective different algorithms in a sparse x setting and provide performance comparisons. Thus, we would like to give brief information related to relation between sparse regularization and LASSO problem.

Original problem of sparse regularization for a k-sparse signal x to be reconstructed is in the form of

$$\min_x ||b - Ax||^2 \;\; s.t. \; ||x||_0 \leq k \tag{3}$$

Main drawback of such an optimization algorithm is that it is a NP-hard problem and best known algorithm to solve it tries all $\binom{n}{k}$ possible supports to find the minimizing x. Idea here is to replace $||x||_0$ by a convex norm by applying convex relaxation. To perform this, we first define the convex envelope.

Definition 1. *A convex envelope of $f(x)$ is the largest convex $q(x)$ such that $q(x) \leq f(x)$, $\forall x \in \mathbb{R}^n$. The convex envelope of a function f is found by computing the Legendre-Fenchel conjugate of f twice.*

$$\begin{aligned} f^*(y) &= \sup_x (y^T x - f(x)) \\ g(x) &= \sup_y (y^T x - f^*(y)) = f^{**}(x) \end{aligned} \tag{4}$$

Theorem 2.1. *The convex envelope of $||x||_0$, when $||x||_\infty \leq 1$ is $||x||_1$.*

Proof: Let $f^*(y) = \sup_{||x||_\infty \leq 1} y^T x - ||x||_0$. Denote the k-th entry of y sorted in non-increasing order as $y_{(k)}$, where $y_{(k)} \geq y_{(k+1)}$. Assume that $||x_0|| = K$. Then, we have $y^T x - ||x||_0 = y^T x - K$. To obtain $\sup y^T x - K$, we can pick x as follows

$$x = \begin{cases} sign(y_i), & \text{if } y_i \text{ is among the largest } K \text{ entries of } y \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

Then, $y^T x - ||x||_0 = \sum_{k=1}^{K} |y_{(k)}| - K = \sum_{k=1}^{K} (|y_{(k)}| - 1)$. Now, if we maximize over K:

$$f^*(y) = \max\{\sum_{k=1}^{N} (|y_{(k)}| - 1) : 0 \leq k \leq N\} = \sum_{k=1}^{N} (|y_{(k)}| - 1)_+ = \sum_{k=1}^{N} (|y_{(k)}| - 1)_+ \tag{6}$$

where $(.)_+$ is the ReLU function. Then,

$$f^{**}(x) = \sup_y y^T x - f^*(y) \tag{7}$$

where x is not restricted. If $\exists\, k \;\, s.t. \;\, |x_k| > 1$, then, for $y = \alpha e_k \, sign(x_k) \implies y^T x - f^*(y) = \alpha |x_k| - (\alpha - 1)_+ \rightarrow \infty$ as $\alpha \rightarrow \infty$, where e_k is the unit one-hot vector with k-th element being 1. Otherwise, when $||x||_\infty \leq 1$, if we pick K such that $|y_{(k)}| > 1$ for $1 \leq k \leq K$,

$$\begin{aligned} y^T x - \sum_{k=1}^{K} (|y_(k)| - 1) &\leq \sum_{k=1}^{N} |y_k||x_k| - ||x||_1 - \sum_{k=1}^{K} (|y_{(k)}| - 1) + ||x||_1 \\ &= \sum_{k=1}^{N} |x_{(k)}|(|y_{(k)}| - 1) - \sum_{k=1}^{K} (|y_{(k)}| - 1) + ||x||_1 \\ &= \sum_{k=1}^{K} (|x_{(k)}| - 1)(|y_{(k)}| - 1) + \sum_{k=K+1}^{N} |x_{(k)}|(|y_{(k)}| - 1) + ||x||_1 \leq ||x||_1 \end{aligned} \tag{8}$$

Equality is reached for $K = 0$, $y_i = sign(x_i)$. Then, $\sum_{k=1}^{K} (|y_k| - 1) = 0 \implies y^T x = ||x||_1$. As a result, convex relaxation of the optimization problem has the form of

$$\min_x ||b - Ax||^2 \;\; s.t. \; ||x||_1 \leq k, \; ||x||_\infty \leq 1 \tag{9}$$

2.3 Unique recovery of x and measurement bound on A

Since in the later parts of the project we will investigate the performance of several different methods on LASSO problem with a sparse x setting, we will briefly provide required properties for unique recovery and bound on m to receive the Restricted Isometry Property (RIP).

Theorem 2.2. *(Unique recovery of x) Unique recovery of a k-sparse $x \in \mathbb{R}^n$ given measurement matrix A is possible if and only if spark(A) $> 2k$, where spark(A) is the minimum number of linearly dependent columns of A.*

We provided the proof in the HW5, and for details please see [5]. Before going further, RIP needs to be defined.

Definition 2. *(RIP) The matrix $A \in \mathbb{R}^{m\times n}$ satisfies the RIP property of order k if $\exists\delta_k \in (0,1)$ such that*

$$(1-\delta_k)||x||_2^2 \leq ||Ax||_2^2 \leq (1+\delta_k)||x||_2^2, \ \ \forall x \in \Sigma_k \tag{10}$$

where Σ_k is the subset of k-sparse signals in $\mathbb{R}^n$.

Theorem 2.3. *(Measurement bound [5]) If $A \in \mathbb{R}^{m\times n}$ satisfies the RIP of order $2k$ with RIP constant $\delta_{2k} \in (0, 1/2]$. Then,*

$$m \geq ck \ln \frac{n}{k} \tag{11}$$

where the constant $c \approx 0.88$.

2.4 LASSO with wavelet transform

In the computational experiments section, we will implement LASSO with wavelet decomposition prior for images and solve the problem with different algorithms. Thus, we very briefly discuss about wavelet transform and why it is useful in LASSO problem for image reconstruction.

The wavelet transform uses functions that are localized in both the real and Fourier space. Generally, the wavelet transform can be expressed by the following equation for a function $f(x)$:

$$F(a,b) = \int_{-\infty}^{\infty} f(x)\psi^*_{(a,b)}(x)\mathrm{d}x \tag{12}$$

where the * is complex conjugate symbol and ψ is the basis functions obeying certain rules including dyadic translations and dilations. Discrete wavelet transform (DWT) is any wavelet transform for which the wavelets are discretely sampled. Most natural images have sparse representations with respect to basis obtained form wavelet transform.

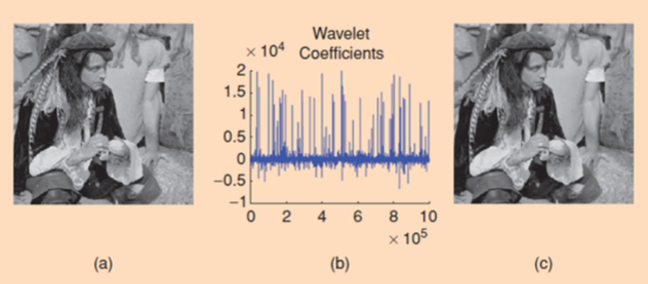


Figure 1: (a) original image containing 10^6 pixels (b) wavelet coefficients in random order (sparse) (c) Reconstructed image using only $25(10^3)$ wavelet coefficients with 97.5% reduction.

In figure 1, this property is illustrated very effectively. Thus, LASSO problem in (1) is modified as follows

$$F(x) \triangleq \frac{1}{2}||b - Ax||^2 + \lambda||Wx||_1 \tag{13}$$

where W is the DWT matrix.

3 Iterative Shrinkage Thresholding Algorithm (ISTA)

3.1 Algorithm Analysis

Iterative Shrinkage Thresholding Algorithm (ISTA) can be traced back to the proximal forward-backward iterative scheme introduced in [2] and [12] within the general framework of splitting methods [6]. The algorithm solves the problem in the following form:

$$\min\{F(x) = f(x) + g(x) : x \in R^n\} \tag{14}$$

The following assumptions are required:

- g: $R^n \to R$ is a continuous convex function which is non-smooth
- f: $R^n \to R$ is a smooth continuously differentiable convex function with Lipschitz continuous gradient $L(f)$: $\|\nabla f(x) - \nabla f(y)\| \geq L(f)\|x - y\|$ for every $x, y \in R^n$

Next, for any L$>$ 0, consider the following quadratic approximation of $F(x)$ at a given point y:

$$Q_L(x, y) = f(y) + \langle x - y, \nabla f(y)\rangle + \frac{L}{2}\|x - y\|^2 + g(x) \tag{15}$$

that has a unique minimizer

$$p_L(y) = \arg\min_{x \in R^n}\{f(y) + \langle x - y, \nabla f(y)\rangle + \frac{L}{2}\|x - y\|^2 + g(x)\} \tag{16}$$

that is by ignoring the constants equal to:

$$p_L(y) = \arg\min_{x \in R^n}\{g(x) + \frac{L}{2}\|x - (y - \frac{1}{L}\nabla f(y))\|^2\} \tag{17}$$

Then the basic step of ISTA is as follows:

$$x_k = p_L(x_{k-1})$$

where 1/L is the step size. In our case $f(x) = \frac{1}{2}\|Ax - b\|_2^2$, $g(x) = \lambda\|x\|_1$. Then ISTA is as follows:

$$x_k = \tau_{\frac{\lambda}{L}}(x_{k-1} - \frac{1}{L}\nabla f(x_{k-1})) = \tau_{\frac{\lambda}{L}}(x_{k-1} - \frac{1}{L}A^T(Ax_{k-1} - b)) \tag{18}$$

where $\tau_{\frac{\lambda}{L}}$ is the soft thresholding operator with threshold at $\frac{\lambda}{L}$.

3.2 Theoretical Convergence Analysis

Theoretical convergence analysis for ISTA can be found in [1]. Generally, for large-scale problems first order methods are the practical option, but it has been observed that they converge slowly, namely they have *sublinear* global rate of convergence. In this section, we will focus on *non-asymptotic* global rate of convergence for ISTA and show that $F(x_k) - F(x^*) \approx O(1/k)$.

Lemma 3.1. *Let f: $R^n \to R$ be a continuously differentiable function with Lipschitz continuous gradient and Lipschitz constant $L(f)$. Then, for any $L \geq L(f)$,*

$$f(x) \leq f(y) + \langle x - y, \nabla f(y)\rangle + \frac{L}{2}\|x - y\|^2 \quad \forall x, y \in R^n \tag{19}$$

Lemma 3.2. *For any $y \in R^n$, one has $z = p_L(y)$ if and only if there exists $\gamma(y) \in \partial g(z)$, the subdifferential of $g(.)$, such that*

$$\nabla f(y) + L(z - y) + \gamma(y) = 0 \tag{20}$$

Lemma 3.3. *Let $y \in R^n$ and $L > 0$ be such that*

$$F(p_L(y)) \leq Q(p_L(y), y) \tag{21}$$

Then for any $x \in R^n$.

$$F(x) - F(p_L(y)) \geq \frac{L}{2}\|p_L(y) - y\|^2 + L\langle y - x, p_L(y) - y\rangle \tag{22}$$

For the proof of Lemma 3.3, please see [1]. (21) directly follows from Lemma 3.1 which is a well-known fundamental property for a smooth function with Lipschitz continuous gradient $L(f)$. Additionally, the proof for Lemma 3.2, is an immediate result from optimality conditions for the strong convex problem. Overall, Lemma 3.1 and Lemma 3.2 are used to obtain Lemma 3.3.

Theorem 3.4. *Let $\{x_k\}$ be the sequence generated by (18). Then for any $k \geq 1$;*

$$F(x_k) - F(x^*) \leq \frac{L(f)\|x_0 - x^*\|^2}{2k} \tag{23}$$

where x^* is any true signal.

Proof: It is basically by invoking Lemma 3.3 with $x = x^*$ and $y = x_n$. Then, we get the following:

$$\frac{2}{L}(F(x^*) - F(x_{n+1})) \geq \|x^* - x_{n+1}\|^2 - \|x^* - x_n\|^2 \tag{24}$$

After summing the inequality over $n = 0, ..., k-1$ and modifying it,we get the desired theorem. For the details please see [1].

In LASSO problem, $L(f) = 2\lambda_{max}(A^TA)$.Then Theorem 3.4 is as follows:

$$F(x_k) - F(x^*) \leq \frac{\lambda_{max}(A^TA)\|x_0 - x^*\|^2}{k} \tag{25}$$

The interpretation for Theorem 3.4 is that error decreases with a rate of $1/k$ where k is the number of iterations. In other words, $F(x_k) - F(x^*) \approx O(1/k)$.

4 A Fast Iterative Shrinkage Thresholding Algorithm (FISTA)

4.1 Algorithm Analysis

The important question is if we can develop a faster method than ISTA described in the previous section while keeping the computational cost the same, i.e. the new method will have the same simplicity of ISTA while its global rate of convergence will be significantly better. This question is answered affirmatively in [1]. Their algorithm for solving (1) is in the following iterative form:

$$x_{k+1} = \tau_{\frac{\lambda}{L}}(p_L(y_k)) \tag{26}$$

where y_k will be chosen smartly by adding "momentum" and p_L is as defined in (16). The algorithm is as follows:

$$\begin{aligned} x_k &= \tau_{\frac{\lambda}{L}}(y_k - \frac{1}{L}A^T(Ay_k - b)) \\ t_{k+1} &= \frac{1 + \sqrt{1 + 4t_k^2}}{2} \\ y_{k+1} &= x_k + \left(\frac{t_k - 1}{t_{k+1}}\right)(x_k - x_{k-1}) \end{aligned}$$

4.2 Theoretical Convergence Analysis

The main difference between ISTA and FISTA is that iterative shrinkage operator is not applied to directly previous x_{k-1} but at a point chosen by using previous two points x_{k-1}, x_{k-2}. Therefore, FISTA is a first-order method which has a *sublinear* global rate of convergence. In this section, we will focus on *non-asymptotic* global rate of convergence for FISTA and show that $F(x_k) - F(x^*) \approx O(1/k^2)$.

Lemma 4.1. *The sequences $\{x_k, y_k\}$ generated using FISTA satisfy for every $k \geq 1$;*

$$\frac{2}{L}t_k^2 v_k - \frac{2}{L}t_{k+1}^2 v_{k+1} \geq \|u_{k+1}\|^2 - \|u_k\|^2 \tag{27}$$

where $v_k = F(x_k) - F(x^*)$, $u_k = t_k x_k - (t_k - 1)x_{k-1} - x^*$

Proof: By invoking Lemma 3.3 with $x = x_k, y = y_{k+1}$ and $x = x^*, y = y_{k+1}$ and applying the usual Pythagoras relation:

$$\|b-a\|^2 + 2\langle b-a, a-c\rangle = \|b-c\|^2 - \|a-c\|^2$$

After some modifications, the desired lemma can be obtained. Please see [1] for the details.

Lemma 4.2. *Let $\{a_k, b_k\}$ be positive sequences of reals satisfying;*

$$a_k - a_{k+1} \geq b_{k+1} - b_k \quad \forall k \geq 1, a_1 + b_1 \leq c, c > 0 \tag{28}$$

Then, $a_k \leq c$ for every $k \geq 1$.

Lemma 4.3. *The positive sequence $\{t_k\}$ generated in FISTA with $t_1 = 1$ satisfies $t_k \geq (k+1)/2$ for all $k \geq 1$.*

Lemma 4.2 and Lemma 4.3 are trivial facts.

Theorem 4.4. *Let $\{x_k\}$, $\{y_k\}$ be generated by FISTA. Then for any $k \geq 1$;*

$$F(x_k) - F(x^*) \leq \frac{2L(f)\|x_0 - x^*\|^2}{(k+1)^2} \tag{29}$$

where x^* is any true signal.

The proof of Theorem 4.4 is a combination of Lemma 4.1, Lemma 4.2, Lemma 4.3 and Lemma 3.3. For the details, you can see [1].

In LASSO problem, $L(f) = 2\lambda_{max}(A^TA)$. Then, Theorem 4.4 is as follows:

$$F(x_k) - F(x^*) \leq \frac{4\lambda_{max}(A^TA)\|x_0 - x^*\|^2}{(k+1)^2} \tag{30}$$

The interpretation for Theorem 3.4 is that error decreases with a rate of $1/k^2$ where k is the number of iterations. In other words, $F(x_k) - F(x^*) \approx O(1/k^2)$.

5 Learned ISTA (LISTA)

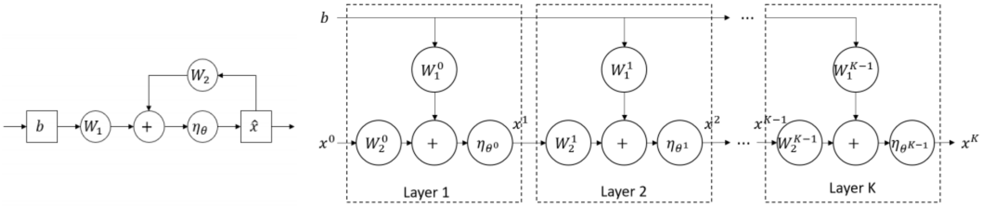
5.1 Algorithm Analysis

Inspired by Iterative Shrinkage Thresholding Algorithm, in [7], they proposed a learning based algorithm named as Learned ISTA (LISTA). The main idea is to train a neural network with a specific architecture that is based on ISTA, to produce the best possible solution to LASSO problem. They unfold ISTA algorithm as a recurrent neural network (RNN) and truncates it into K iterations. In Figure 2 (a), recurrent structure of ISTA can be seen where $W_1 = \frac{1}{L}A^T$, $W_2 = I - \frac{1}{L}A^TA$ and η_θ is the soft thresholding function with threshold at θ. In Figure 2 (b) unfolded structure of LISTA can be seen where all the parameters $\{(W_1^k, W_2^k, \theta^k)\}_{k=0}^{K-1}$ are learned by stochastic gradient descent (SGD) over a given a given training dataset $\{(x_i^*, b_i)\}_{i=1}^{P}$. Unfolded structure of LISTA can be summarized as follows:

$$x^{k+1} = \eta_{\theta^k}(W_1^k b + W_2^k x^k), \quad k = 0, ..., K-1 \tag{31}$$

The training loss function is as follows:

$$\min_{\{(W_1^k, W_2^k, \theta^k)\}_{k=0}^{K-1}} \frac{1}{P}\sum_{p=0}^{P-1} \frac{1}{2}\|x^K\left(W_1^k, W_2^k, \theta^k, b_i\right) - x_i^*\|_2^2 \tag{32}$$



(a) RNN structure of ISTA.

(b) Unfolded learned ISTA Network.

Figure 2: (a) ISTA, (b) LISTA

5.2 Theoretical Convergence Analysis

The first attempt to establish a theoretical convergence rate of LISTA is done in [4]. The details of the convergence analysis in this section can be found in [4].

Let's define the following noisy linear measurements:

$$b = Ax^* + w \tag{33}$$

Assumption 1. *Signal x^* and observation noise w are sampled from the following set:*

$$(x^*, w) \in X(B, s, \sigma) = \{(x^*, w) : |x_i^*| \leq B, \forall i, \|x^*\|_0 \leq s, \|w\|_1 \leq \sigma\} \tag{34}$$

Assumption 1 is composed of basic assumptions that implies x^* is bounded and s-sparse($s \geq 2$) and w is bounded.

Lemma 5.1 (Necessary Condition)**.** *Given $\{(W_1^k, W_2^k, \theta^k)\}_{k=0}^{\infty}$ and x^0, let b be observed by (33) and $\{x_k\}_{k=1}^{\infty}$. If the following holds uniformly for any $(x^*, w) \in X(B, s, 0)$(no noise):*

$$x^k(\{(W_1^k, W_2^k, \theta^k)\}_{k=0}^{\infty}, b, x^0) \to x^*, as \quad k \to \infty$$

and $\{W_2^k\}_{k=1}^{\infty}$ are bounded

$$\|W_2^k\|_2 \leq B_W, \forall k = 0, 1, 2, ...$$

then $\{W_1^k, W_2^k, \theta^k\}_{k=0}^{\infty}$ must satisfy

$$W_2^k - (I - W_1^k A) \to 0, as \quad k \to \infty \tag{35}$$

$$\theta^k \to 0, as \quad k \to \infty \tag{36}$$

The result of equation (35), in LISTA $\{W_1^k, W_2^k\}$ asymptotically satisfies $W_2^k = I - W_1^k A$. In [4], they adopt that and simplify the LISTA to:

$$x^{k+1} = \eta_{\theta^k}(x^k + (W^k)^T(b - Ax^k)), \quad k = 0, ..., K-1 \tag{37}$$

where $\{W^k, \theta^k\}_{k=0}^{K-1}$ are trainable parameter This method is named as LISTA-CP where CP stands for partial weight coupling.

Theorem 5.2 (Convergence of LISTA-CP)**.** *Given $\{W^k, \theta^k\}_{k=0}^{\infty}$ and $x^0 = 0$, let $\{x^k\}_{k=1}^{\infty}$ be generated by (37). If Assumption 1 holds and s is sufficiently small, then there exists a sequence of parameters $\{W^\tau, \theta^\tau\}_{\tau=0}^{k-1}$ such that, for all $(x^*, w) \in X(B, s, \sigma)$, we have the error bound:*

$$\|x^k(\{W^\tau, \theta^\tau\}_{\tau=0}^{k-1}, b, x^0) - x^*\|_2 \leq sB\,exp(-ck) + C\sigma, \forall k = 1, 2, ... \tag{38}$$

where $c > 0, C > 0$ are constants depend only on A and s.

Investigating noiseless case by choosing ($\sigma = 0$), the convergence rate becomes as follows:

$$\|x^k - x^*\|_2 \leq sB\,exp(-ck) \tag{39}$$

where $s < (1 + 1/\mu)/2$ and $c = -log(2\mu s - \mu)$. μ is defined as *"Generalized Mutual Coherence"*:

$$\mu(A) = \inf_{W \in R^{m \times n}, (W_i)^T A_i = 1, 1 \leq i \leq n} \{ \max_{i \neq j, 1 \leq i, j \leq n} |W_i^T A_j| \} \tag{40}$$

where A_i and W_i are unit norm vectors. One observation from Theorem 5.2 combined with Lemma 5.1, is that partial weight coupling is both necessary and sufficient to guarantee convergence.

6 Alternating Direction Method of Multipliers (ADMM)

6.1 Algorithm Analysis

Alternating Direction Method of Multipliers is a decomposition-coordination procedure, where solutions to small subproblems are coordinated to solve the global problem. It utilizes the dual decomposition and augmented Lagrangian method for constrained optimization. Algorithm uses, dual ascent, dual decomposition, augmented Lagrangian and the Method of Multipliers.

Algorithm solves the problem of the form

$$\min f(x) + g(z) \;\; s.t. \;\; Kx + Nz = c \tag{41}$$

where $x \in \mathbb{R}^n$, $z \in \mathbb{R}^m$, $A \in \mathbb{R}^{p \times n}$, $B \in \mathbb{R}^{p \times m}$ and $c \in \mathbb{R}^p$. f and g are convex. The initial variable x is divided into two parts, called x and z in this context, where the objective function is separated across this splitting. The optimal value of the problem 41 is obtained by

$$p^* = \inf\{f(x) + g(z) | Kx + Nz = c\} \tag{42}$$

In order to increase the robustness of the dual ascent method and obtain convergence without strict convexity or finiteness assumptions on f, the augmented Lagrangian is formed:

$$L_\rho(x, z, y) = f(x) + g(z) + y^T(Kx + Nz - c) + (\rho/2)||Kx + Nz - c||_2^2 \tag{43}$$

where $\rho > 0$ is the penalty parameter. This problem is equivalent to the original problem (41), any feasible x making this term 0. Benefit of including this term is that it makes the associated dual function $g_\rho(y) = \inf_{x,z} L_\rho(x, z, y)$ differentiable under less strict conditions on the original problem.

Ultimately, ADMM iterations can be stated as

$$\begin{aligned} x^{k+1} &= arg \min_x L_\rho(x, z^k, y^k) \;\; \text{(x-minimization step)} \\ z^{k+1} &= arg \min_x L_\rho(x^{k+1}, z, y^k) \;\; \text{(z-minimization step)} \\ y^{k+1} &= y^k + \rho(Kx^{k+1} + Nz^{k+1} - c) \quad \text{(dual variable update)} \end{aligned} \tag{44}$$

where $\rho > 0$. Dual variable update stepsize is equal to the augmented Lagrangian parameter ρ.

The method of multipliers for (41) has the form

$$\begin{aligned} (x^{k+1}, z^{k+1}) &= arg \min_{x,z} L_\rho(x, z, y^k) \\ y^{k+1} &= y^k + \rho(Kx^{k+1} Nz^{k+1} - c) \end{aligned} \tag{45}$$

Difference in ADMM is that, x and z are updated in an alternating scheme rather than simultaneously. Separating the minimization over x and z into two steps allows for decomposition when f and g are separable.

6.1.1 Scaled Form of ADMM

If we combine linear and quadratic terms in the augmented Lagrangian and scale the dual variable, we obtain

$$\begin{aligned} y^T + (\rho/2)||r||_2^2 &= (\rho/2)||r + (1/\rho)y||_2^2 - (1/2\rho)||y||_2^2 \\ &= (\rho/2)||r + u||_2^2 - (\rho/2)||u||_2^2 \end{aligned} \tag{46}$$

where $r = Kx + Nz - c$ is the residual and $u = (1/\rho)y$ is the dual variable. Then, ADMM iterations have the form of

$$\begin{aligned} x^{k+1} &= arg\min_x(f(x) + (\rho/2)||Kx + Nz^k + u^k - c||_2^2) \\ z^{k+1} &= arg\min_x(g(z) + (\rho/2)||Kx^{k+1} + Nz + u^k - c||_2^2) \\ u^{k+1} &= u^k + Kx^{k+1} + Nz^{k+1} - c \end{aligned} \tag{47}$$

If we define the residual at iteration k as $r^k = Kx^k + Nz^k - c$, u^k is equal to the running sum of the residuals

$$u^k = u^0 + \sum_{j=1}^{k} r^j \tag{48}$$

6.1.2 ADMM application for Generalized LASSO

The function to be optimized for Generalized LASSO problem has the form of

$$\min_x \frac{1}{2}||Ax - b||_2^2 + \lambda||Fx||_1 \tag{49}$$

where $\lambda > 0$ is usually chosen by cross-validation and F is an arbitrary linear transformation. In ADMM form, the LASSO problem can be stated as follows

$$\min_{x,z} f(x) + g(z) \;\; s.t. \;\; x - z = 0 \tag{50}$$

where $f(x) = \frac{1}{2}||b - Ax||_2^2$ and $g(z) = \lambda||Fz||_1$. Consequently, iterations for ADMM have the form of

$$\begin{aligned} x^{k+1} &= (A^TA + \rho F^TF)^{-1}(A^Tb + \rho F^T(z^k - u^k)) \\ z^{k+1} &= S_{\lambda/\rho}(Fx^{k+1} + u^k) \\ u^{k+1} &= u^k + Fx^{k+1} - z^{k+1} \end{aligned} \tag{51}$$

where $(A^TA + \rho I)$ is always invertible due to $\rho > 0$ if F is a unitary transform where $F^TF = I$ and the x-update is a ridge regression.

In the empirical comparison of algorithms part, two different problem settings will be considered for F.

6.1.3 Theoretical Convergence Rate Analysis

In this section, a general lower bound on the convergence rate is obtained for general form of over-relaxed ADMM, which is a larger set of algorithms that includes the ADMM itself. However, this analysis requires the $f(x)$ to be strongly convex. Thus, computational analysis of this part is performed with a specific setting.

Let $S_d(m, L)$ denote the set of differentiable convex functions $f : \mathbb{R}^d \to \bar{\mathbb{R}}$ that are strongly convex with parameter m and Lipschitz continuous with parameter L. Also, let $S_d(0, \infty)$ denote the set of convex functions $\mathbb{R}^d \to \bar{\mathbb{R}}$.

Assumption 2. *f and g are closed, proper and convex, which is coherent with the LASSO problem. Assume that for some $0 < m \leq L < \infty$ we have $f \in S_p(m, L)$, $g \in S_q(0, \infty)$, A is invertible and B has full column rank. These assumptions hold for LASSO problem since $K = N = I_p$ and f is strongly convex.*

Derivations [10] are done for over-relaxed version of ADMM where each instance of Kx_{k+1} in the z and u updates is replaced by

$$\alpha Kx_{k+1} - (1 - \alpha)(Nz_k - c)$$

If we select $\alpha = 1$, we obtain ADMM itself.

Theorem 6.1. *[8] Suppose that assumption above holds. Let the sequences x_k, z_k and u_k be generated by running over relaxed ADMM algorithm with step size $\rho = (\hat{m}\hat{L})^{\frac{1}{2}}\rho_0$ and over relaxation parameter α, where $\hat{m} = \frac{m}{\sigma_1^2(K)}$, $\hat{L} = \frac{L}{\sigma_p^2(K)}$, and $\kappa = \kappa_f\kappa_K^2 = \hat{L}/\hat{m}$ where, $\kappa_f = \frac{L}{m}$, $\kappa_K = \frac{\sigma_1(K)}{\sigma_p(K)}$, σ_1 and σ_p are the* *largest and smallest singular values of K. Suppose (x_*, z_*, u_*) is a fixed point of algorithm 2, and define $\varphi_k = \begin{bmatrix} z_k \\ u_k \end{bmatrix}$, $\varphi_* = \begin{bmatrix} z_* \\ u_* \end{bmatrix}$. Fix $0 < \tau < 1$, and suppose that a 2x2 matrix $P \succ 0$ and $\lambda^1, \lambda^2 \geq 0$ such that the 4x4 linear matrix inequality*

$$\begin{aligned} 0 \succeq & \begin{bmatrix} \hat{A}^\top P \hat{A} - \tau^2 P & \hat{A}^\top P \hat{B} \\ \hat{B}^\top P \hat{A} & \hat{B}^\top P \hat{B} \end{bmatrix} \\ & + \begin{bmatrix} \hat{C}^1 & \hat{D}^1 \\ \hat{C}^2 & \hat{D}^2 \end{bmatrix}^\top \begin{bmatrix} \lambda^1 M^1 & 0 \\ 0 & \lambda^2 M^2 \end{bmatrix} \begin{bmatrix} \hat{C}^1 & \hat{D}^1 \\ \hat{C}^2 & \hat{D}^2 \end{bmatrix} \end{aligned} \tag{52}$$

is satisfied, where $\hat{A} = \begin{bmatrix} 1 & \alpha - 1 \\ 0 & 0 \end{bmatrix}$, $\hat{B} = \begin{bmatrix} \alpha & -1 \\ 0 & -1 \end{bmatrix}$, $\hat{C}^1 = \begin{bmatrix} -1 & -1 \\ 0 & 0 \end{bmatrix}$, $\hat{D}^1 = \begin{bmatrix} -1 & 0 \\ 1 & 0 \end{bmatrix}$, $\hat{C}^2 = \begin{bmatrix} 1 & \alpha - 1 \\ 0 & 0 \end{bmatrix}$, $\hat{D}^2 = \begin{bmatrix} \alpha & -1 \\ 0 & 1 \end{bmatrix}$ *and* $M^1 = \begin{bmatrix} -2\rho_0^{-2} & \rho_0^{-1}\left(\kappa^{-\frac{1}{2}} + \kappa^{\frac{1}{2}}\right) \\ \rho_0^{-1}\left(\kappa^{-\frac{1}{2}} + \kappa^{\frac{1}{2}}\right) & -2 \end{bmatrix}$, $M^2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$

Then, for all $k \geq 0$, we have

$$\|\varphi_k - \varphi_*\| \leq \kappa_N \sqrt{\kappa_P}\, \|\varphi_0 - \varphi_*\|\, \tau^k$$

Detailed proof of the theorem can be found in the appendix of[10]. For fixed values of $\alpha, \rho_0, \hat{m}, \hat{L}$ and τ, (52) is a semidefinite program with variables P, λ^1 and λ^2. Minimal rate of τ is found via binary search such that the linear matrix inequality in (52) is satisfied. Note that when $\rho_0 = \kappa^\epsilon$, the matrix $M^1 = \begin{bmatrix} -2\kappa^{-2\varepsilon} & \kappa^{-\frac{1}{2}-\varepsilon} + \kappa^{\frac{1}{2}-\varepsilon} \\ \kappa^{-\frac{1}{2}-\varepsilon} + \kappa^{\frac{1}{2}-\varepsilon} & -2 \end{bmatrix}$ and (52) only depends on κ.

One caveat about Theorem 6.1 is that convergence rate is not given as a function of κ. To use it, κ is specified beforehand and search is done for the minimal τ s.t. (52) is feasible. This problem is solved by obtaining the rate as a symbolic function of the step size ρ and the over-relaxation parameter α.

Theorem 6.2. *Suppose again the assumption holds. Let the sequences x_k, z_k and u_k be generated by running over-relaxed ADMM algorithm with parameter $\alpha \in (0, 2)$, with step size $\rho = (\hat{m}\hat{L})^{1/2}\kappa^\epsilon$, where $\epsilon \in (-\infty, \infty)$. Define x_*, z_*, u_*, φ_k and φ_* as in Theorem 3.1. Then, for all sufficiently large κ, we have*

$$\|\varphi_k - \varphi_*\| \leq C\, \|\varphi_0 - \varphi_*\| \left(1 - \frac{\alpha}{2\kappa^{0.5+|\varepsilon|}}\right)^k \tag{53}$$

where $C = \kappa_B \sqrt{\max\left\{\frac{\alpha}{2-\alpha}, \frac{2-\alpha}{\alpha}\right\}}$.

Detailed proof is included in the appendix.

For the LASSO problem solved by ADMM, which is the case that we are interested in, convergence rate result in Theorem 6.2 has the form of

$$\|\varphi_k - \varphi_*\| \leq \|\varphi_0 - \varphi_*\| \left(1 - \frac{1}{2\kappa^{0.5}}\right)^k \tag{54}$$

where $\alpha = 1$, $K = N = I_p$, $C = 1$, $\kappa_K = \kappa_N = 1$, $\kappa_f = L_f/m_f$ and $\epsilon = 0$ is selected. m_f and L_f can be found as the smallest and largest eigenvalues of $A^T A$, respectively.

7 Plug and Play (PnP) Priors for Model Based Reconstruction

7.1 Algorithm Analysis

Plug and Play Priors for Model Based Reconstruction [13] (PnP) provides a flexible network for using denoising algorithms as priors for model-based inversion. It simplifies the software implementation and integration. Also, it allows denoising methods that are not explicitly formulated as optimization problems to be utilized inside a direct application of ADMM, which solves various MAP regularized inverse problems. In PnP, prior and forward model terms of MAP estimation problem is decoupled by splitting the state variable, resulting in two decoupled optimization problems, one for the forward model and one for the prior model. Decoupled software implementation becomes possible after splitting.

7.1.1 MAP cost function for solving inverse problems

Let y be our $M \times 1$ measurement vector, $p(b|x)$ be the conditional probability density function (pdf) of the measurements b given x and $p(x)$ be the pdf of the unknown. MAP estimate of x has the form of

$$\begin{aligned} \hat{x} &\leftarrow \underset{x}{\operatorname{argmin}}\{-\log p(b|x) - \log p(x)\} \\ \hat{x} &\leftarrow \underset{x}{\operatorname{argmin}}\{l(b;x) + s(x)\} \end{aligned} \tag{55}$$

where $l(b;x) = -\log p(b|x)$ and $-\log p(x) = s(x)+$ terms independent of x. When $l(b;x) = \frac{1}{2\sigma_n^2}\|b-x\|_2^2 + \frac{M}{2}\log\left(2\pi\sigma_n^2\right)$, MAP estimate tailored to remove the white Gaussian noise with variance σ_n^2. The operator that denoises b with $n \sim N(0,\sigma_n^2)$ is given by:

$$\mathbb{H}\left(b;\sigma_n^2\right) = \underset{x}{\operatorname{argmin}}\left\{\frac{1}{2\sigma_n^2}\|b-x\|_2^2 + s(x)\right\} \tag{56}$$

To control relative effect of the prior model on the reconstruction, problem can be rewritten as

$$\hat{x} = \underset{x}{\operatorname{argmin}}\{l(b;x) + \lambda s(x)\} \tag{57}$$

where $\lambda = 1$ gives exact MAP estimation problem. To implement splitting on this problem, we can reformulate it as

$$\hat{x} = \underset{x}{\operatorname{argmin}}\{l(b;x) + \lambda s(z)\} \;\; s.t. \; x = z \tag{58}$$

Then, as shown in the previous sections, we can form augmented Lagrangian, L_ρ:

$$L_\rho(x,z,u) = l(b;x) + \lambda s(z) + \frac{\rho}{2}\|x - z + u\|_2^2 - \frac{\rho}{2}\|u\|_2^2 \tag{59}$$

and use ADMM steps at k-th iteration step for the solution as

$$\begin{aligned} x_k &= \underset{x}{\operatorname{argmin}} L_\rho(x, z_{k-1}, u_{k-1}) \\ z_k &= \underset{z}{\operatorname{argmin}} L_\rho(x_k, z, u_{k-1}) \\ u_k &= u_{k-1} + (x_k - z_k) \end{aligned} \tag{60}$$

If $\tilde{x} = z_k - u_{k-1}$ and $\tilde{z} = x_k + u_{k-1}$, ADMM steps have the form of

$$\begin{aligned} x_k &= \mathbb{F}(b,\tilde{x};\rho) = \underset{x}{\operatorname{argmin}}\left\{l(b;x) + \frac{\rho}{2}\|x - \tilde{x}\|_2^2\right\} \\ z_k &= \mathbb{H}(\tilde{z};\frac{\lambda}{\rho}) = \underset{z}{\operatorname{argmin}}\left\{\frac{\rho}{2}\|\tilde{z} - z\|_2^2 + \lambda s(z)\right\} \\ u_k &= u_{k-1} + (x_k - z_k) \end{aligned} \tag{61}$$

The first step of the algorithm only depends on the forward model and returns the MAP estimate of x given b. The second step only depends on the choice of prior and can be interpreted as a denoising operation as in [56]. This framework, allows the use of two independent modules, one for $\mathbb{F}$ and one for $\mathbb{H}$ respectively. Changing the prior model only modifies H. Hence, PnP framework allows the utilization of different prior models, $\mathbb{H}$, with the forward model $\mathbb{F}$ of interest. The minimization corresponding to $\mathbb{F}$ and $\mathbb{H}$ does not need to be exact. They can be replaced by approximate operators that do not minimize the cost functions but decrease it sufficiently. By this way, speed of ADMM can be enhanced for practical purposes.

7.2 Theoretical Convergence Analysis

Since convergence rate and global convergence is specific to the denoiser type, only the fixed-point convergence result for PnP algorithms is included. To ensure global convergence for PnP ADMM, a sufficient condition is having denoising algorithm $\mathbb{H}_\gamma$ $\gamma = \lambda/\rho_k$, where ρ_k is the augmented Lagrangian coefficient at k-th step, has symmetric gradient and is non-expansive. In this case, g exists due to a proximal mapping theorem [9]. However, proving non-expansive denoisers can be difficult. Thus, we are interested in fixed-point convergence. Fixed point convergence guarantees that a nonlinear algorithm can enter into a steady state asymptotically and for any initial point lying in a region called the basin of attraction it will converge. To show fixed-point convergence [3], bounded denoisers are defined first.

Definition 3. *(Bounded Denoiser) A bounded denoiser with a parameter γ is a function $\mathbb{H}_\gamma : \mathbb{R}^n \to \mathbb{R}^n$ such that $\forall x \in \mathbb{R}^n$,*

$$\left\| \mathbb{H}_\gamma(\boldsymbol{x}) - \boldsymbol{x} \right\|^2 / n \leq \gamma^2 C \tag{62}$$

for some constant C independent of n and γ. We have $\mathbb{H}_\gamma \to I$ as $\gamma \to 0$.

Assumption 3. *$f : [0,1]^n \to \mathbb{R}$ has bounded gradients. $\forall x \in [0,1]^n$, $\exists\, L < \infty$ such that $||\nabla f(x)||_2/\sqrt{n} \leq L$.*

Also, for $f(x) = ||Ax - b||_2^2$, $A \in R^{n \times n}$ with eigenvalues in $[0,1]$, we have $\nabla f(x) = 2A^T(Ax - b)$ and

$$\|\nabla f(\boldsymbol{x})\|_2/\sqrt{n} \leq 2\lambda_{\max}(\boldsymbol{A})\left(\|\boldsymbol{x}\|_2 + \|\boldsymbol{b}\|_2\right)/\sqrt{n}$$

Then, main convergence result is as follows.

Theorem 7.1. *Under Assumption 3, for any $\mathbb{H}_\gamma$, the iterates of PnP ADMM demonstrates a fixed-point convergence. Namely, $\exists\,(x^*, z^*, u^*)$ such that $||x_k - x^*||_2 \to 0$, $||z_k - z^*||_2 \to 0$ and $||u_k - u^*||_2 \to 0$ as $k \to \infty$.*

Proof of the theorem is included in [3]. Theorem 7.1 ensures that $x_k \to z_k$ since $u_{k+1} = u_k + (x_{k+1} - z_{k+1})$ converges. A reasonable stopping criterion can be selected as

$$\begin{aligned}\Delta_{k+1} = \frac{1}{\sqrt{n}} \Big(\left\| \boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)} \right\|_2 + \left\| \boldsymbol{v}^{(k+1)} - \boldsymbol{v}^{(k)} \right\|_2 \\ + \left\| \boldsymbol{u}^{(k+1)} - \boldsymbol{u}^{(k)} \right\|_2 \Big) \leq \text{tol}\end{aligned} \tag{63}$$

for some tolerance level tol.

8 Computational Costs of Algorithms

In this section, we will compare computational costs of ISTA, FISTA, LISTA, ADMM and Plug and Play. We will first calculate cost per iteration and then we will investigate the cost of obtaining ϵ approximate solution. First of all, let's define a generic matrix $A \in R^{m \times n}$ as in Equation (1).

The computation cost of ISTA per iteration comes from mainly matrix vector multiplication that is in $A^T A x_{k-1}$ form where $A^T A \in R^{n \times n}$ and $x_{k-1} \in R^n$. Then, computational complexity of ISTA per iteration is $O(n^2)$.

Similarly, the computation cost of FISTA per iterations comes from mainly matrix vector multiplication that is in $A^T A y_k$ form where $A^T A \in R^{n \times n}$ and $y_k \in R^n$. Then, computational complexity of FISTA per iteration is $O(n^2)$.

The computation cost of LISTA per iteration is exactly same as the computation cost of ISTA. The main cost comes from matrix vector multiplication of $W_2^k x^k$ that is $O(n^2)$.

The computation cost of ADMM per iteration comes mainly from matrix inversion($(A^T A + \rho I)^{-1}$) and matrix vector multiplication($(A^T A + \rho I)^{-1}(z^k - u^k)$). The cost of matrix inversion is $O(n^3)$ when we use Gauss-Jordan elimination and the cost of matrix vector product is $O(n^2)$.

For Plug and Play Algorithm, it depends on the denoiser algorithm. The first step still is dominated by matrix inversion which is $O(n^3)$. If the computational cost of denoiser algorithm is heavier that $O(n^3)$, then computational cost of PnP is dominated by denoiser step. Otherwise, PnP is dominated by $O(n^3)$.

A comparative analysis of computational costs for different algorithms can be found in the following Table 1.

	ISTA	FISTA	LISTA	ADMM
Cost per Iteration	$O(n^2)$	$O(n^2)$	$O(n^2)$	$O(n^3)$
Number of Iterations to achieve ϵ	$O(\frac{1}{\epsilon})$	$O(\frac{1}{\sqrt{\epsilon}})$	$O(ln(\frac{1}{\epsilon}))$	$O(log_\beta(\frac{\epsilon}{c}))$
Total Cost	$O(\frac{n^2}{\epsilon})$	$O(\frac{n^2}{\sqrt{\epsilon}})$	$O(n^2 ln(\frac{1}{\epsilon}))$	$O(n^3 log_\beta(\frac{\epsilon}{c}))$

Table 1: Comparative Analysis of Computational Costs of Different Algorithms for LASSO problem

where c and β can be defined by using (54).

9 Computational Experiments

9.1 Sparse recovery: Measurement Bound Experiments with LASSO

In order to demonstrate the relation between the sparse recovery of x and the number measurements taken from it, we conducted several experiments with different configurations since we are going to compare the performances of the algorithms on sparse recovery problem.

9.1.1 Robust CS problem - Noiseless measurement - Random Gaussian Measurement Matrix RIP condition analysis

In this experiment, we attempted to demonstrate the validity of the theoretical measurement bound result in (11) for which the measurement matrix A in (1) satisfies RIP of order $2k$ with RIP constant $\delta_{2k} \in (0, \frac{1}{2}]$. For this purpose, we generated a measurement matrix A where every element $A_{ij} \sim N(0, 1/m)$, since such matrices are shown to satisfy RIP with exponentially high probability with number of measurements, m, as in [11]. For the experiments, sparsity level of $x \in \mathbb{R}^n$ was selected exactly as $k = 6$ and $n = 200$. Using this, we obtained the lower bound on measurements to satisfy RIP as $m \geq 18.67$ using (11). Then, we initialized the measurement matrix with $m = 10$ and added one more column at each step, swept over the λ in (1) and total number of iterations using ADMM to solve (1) to find the minimum $||x - \tilde{x}||_2$ at each step, where $\tilde{x}$ is the reconstruction. Then, we obtained the plot which indicates the minimum $l2$ reconstruction error versus the number of measurements using ADMM algorithm.

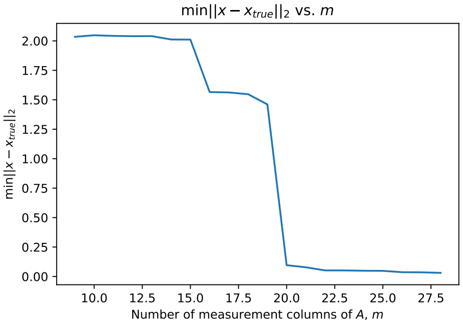


Figure 3: $l2$ reconstruction error vs. number of measurement columns, m, where $m \in \{10, .., 35\}$

Result of this analytical study was coherent with the theoretical results. After $m = 19$, fast drop in the reconstruction error can be seen from the plot itself.

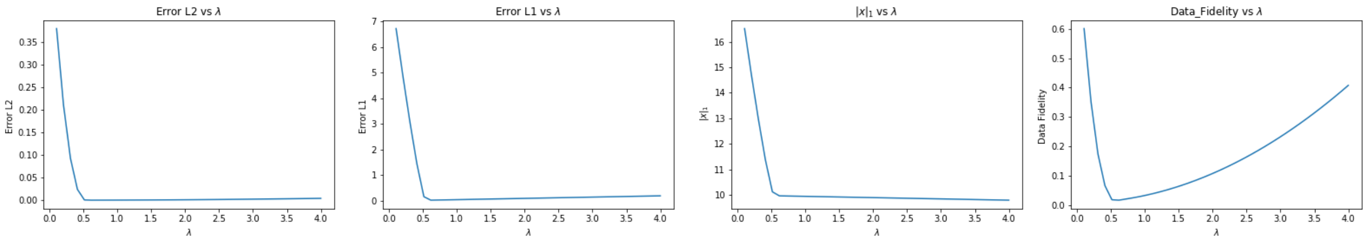
9.2 Empirical Analysis of the Theoretical Convergence Rate/Analysis for provided algorithms for LASSO problem

9.2.1 Selection of λ

λ is very critical for solution of Problem (1). It determines the trade-off between data fidelity term which is the least squares term, and prior knowledge which is the sparsity or l_1 term. High level intuition is that if we have simpler problem then we need to pick smaller λ. E.g. When the noise power is small we can rely on the data fidelity term more and make λ smaller or when the forward problem has smaller condition number then we can choose λ smaller again.

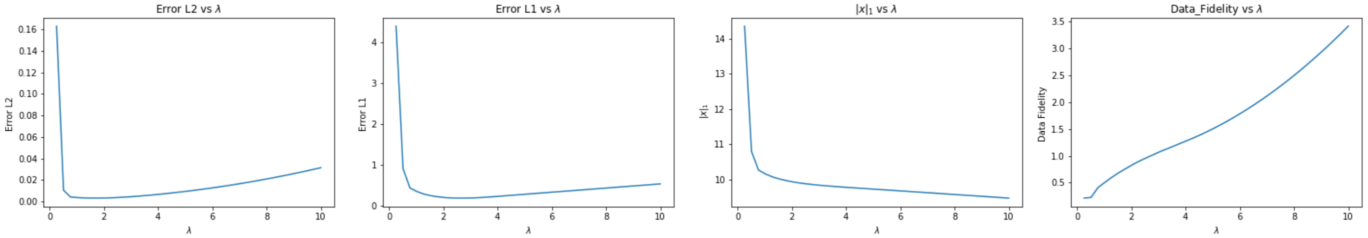
In our experiments, we selected λ empirically. Firstly, we generated a validation set that consists of 100 true signals. Based on our validation set, we obtain 100 measurements. Similarly we defined a possible set of λs. Then the problem (1) was solved for each elements in the set of λs. Subsequently, we obtained four different graphs:(i) $\|x_{true} - x_{recovered}\|_2$ vs λ, (ii) $\|x_{true} - x_{recovered}\|_1$ vs λ, (iii) $\|x_{recovered}\|_1$ vs λ , (iv) $\|Ax_{recovered} - y\|_2$ vs λ. Then, λ selection can be done by inspecting those 4 graphs.

In the following figures, we have investigated two different scenarios to clarify λ selection. In both cases, the condition number of the forward operator was 20 and sparsity level was chosen as 10. In figure 4, noise variance is 0.01, in figure 5, noise variance is 0.1. In the first case, the optimal λ is around 0.5. On the other hand, in the second case, the optimal λ is around 2. As the noise level increases, it is legitimate to choose λ larger.



(a) (b) (c) (d)

Figure 4: (a)$\|x_t - x_r\|_2$ vs λ, (b)$\|x_{true} - x_r\|_1$ vs λ, (c)$\|x_r\|_1$ vs λ, (d)$\|Ax_r - y\|_2$ vs λ



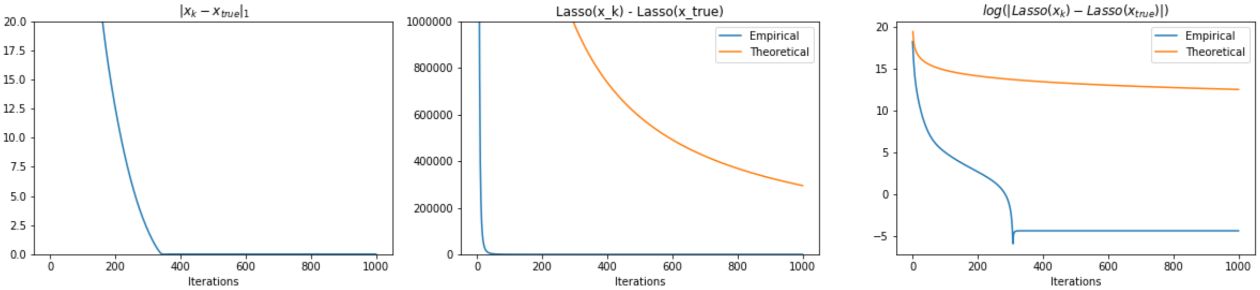
(a) (b) (c) (d)

Figure 5: (a)$\|x_t - x_r\|_2$ vs λ, (b)$\|x_{true} - x_r\|_1$ vs λ, (c)$\|x_r\|_1$ vs λ, (d)$\|Ax_r - y\|_2$ vs λ

9.2.2 ISTA

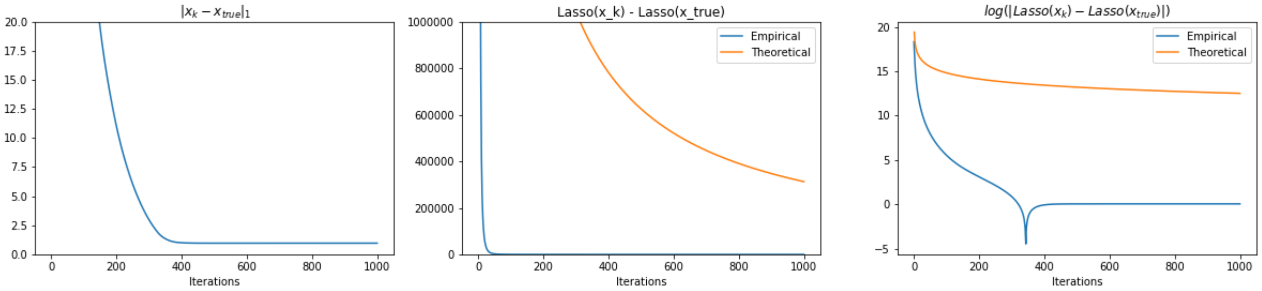
In section 3.2, we provided the theoretical convergence rate analysis for ISTA as a general lower bound. The analysis was regarding *non-asymptotic* global rate of convergence for ISTA and it was shown that $F(x_k) - F(x^*) \approx O(1/k)$ that can be found in (25).

In the experiments, we worked with a square measurement matrix $A \in \mathbb{R}^{n \times n}$. Matrix A is obtained by randomly sampling a Gaussian distribution where $A_{ij} \sim N(0,1)$, $i,j \in \{1,...,n\}$. $x \in \mathbb{R}^n$, where $n = 200$, was selected as 5-sparse where nonzero elements were 1. In the experiments, the condition number of matrix A is 130. Then, results for $F(x_k) - F(x^*)$ are obtained and compared with the theoretical bounded that is provided in (25). Respective lower bound on the rate was verified for measurements with and without noise for two cases that can be seen in Figure (6) and (7).



(a) (b) (c)

Figure 6: Convergence rate analysis for ISTA algorithm with measurements y without noise ($\lambda = 1$). (a) Reconstruction error, (b) Linear scale convergence rate and lower bound on the rate , (c) Log scale convergence rate and lower bound on the rate.



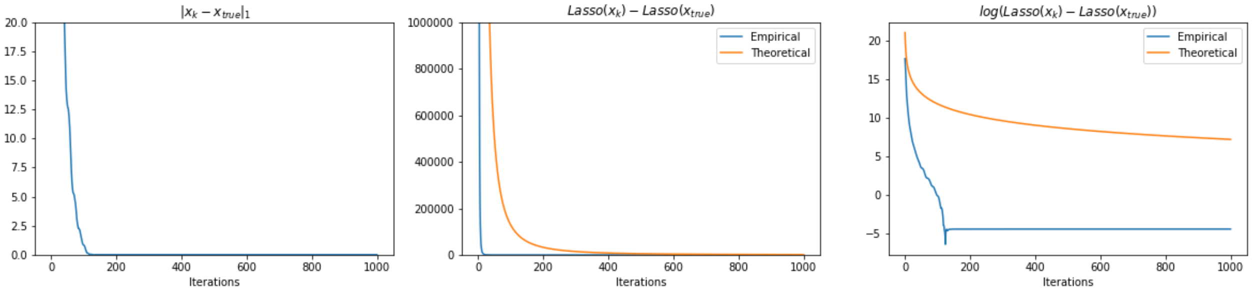
(a) (b) (c)

Figure 7: Convergence rate analysis for ISTA algorithm with measurements y with Gaussian noise $N(0, 0.2)$ ($\lambda = 1$). (a) Reconstruction error, (b) Linear scale convergence rate and lower bound on the rate, (c) Log scale convergence rate and lower bound on the rate.

9.2.3 FISTA

In section 4.2, we provided the theoretical convergence rate analysis for FISTA as a general lower bound. The analysis was regarding *non-asymptotic* global rate of convergence for ISTA and it was shown that $F(x_k) - F(x^*) \approx O(1/k^2)$ that can be found in (30).

In the experiments, we worked with a square measurement matrix $A \in \mathbb{R}^{n\times n}$. Matrix A is obtained by randomly sampling a Gaussian distribution where $A_{ij} \sim N(0, 1)$, $i, j \in \{1, ..., n\}$. $x \in \mathbb{R}^n$, where $n = 200$, was selected as 5-sparse where nonzero elements were 1. In the experiments, the condition number of matrix A is 130. Then, results for $F(x_k) - F(x^*)$ are obtained and compared with the theoretical bounded that is provided in (30). Respective lower bound on the rate was verified for measurements with and without noise for two cases that can be seen in Figure (8) and (9).



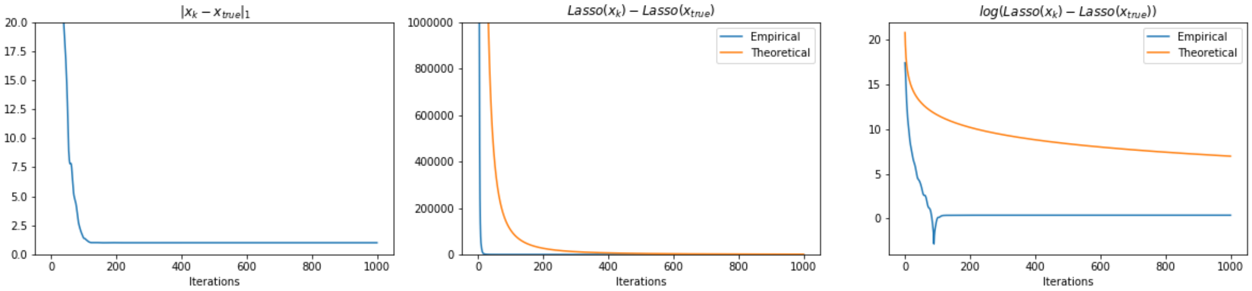
(a) (b) (c)

Figure 8: Convergence rate analysis for FISTA algorithm with measurements y without noise ($\lambda = 1$). (a) Reconstruction error, (b) Linear scale convergence rate and lower bound on the rate, (c) Log scale convergence rate and lower bound on the rate.

9.2.4 LISTA

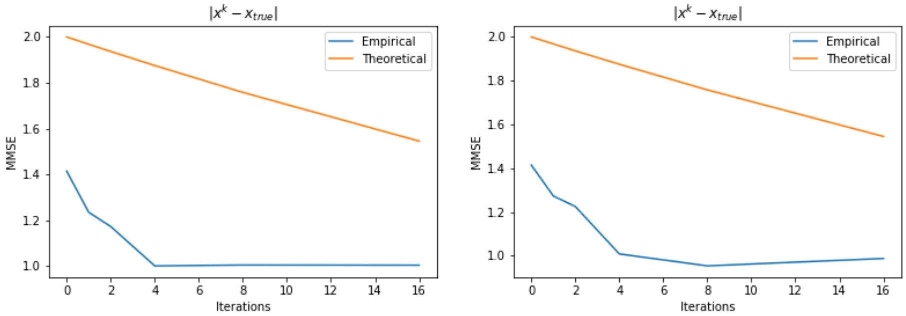
In section 5.2, we provided the theoretical convergence rate analysis for LISTA as a general lower bound. The analysis was regarding *asymptotic* global rate of convergence for LISTA and it was shown that $\|x_k - x^*\|_2 \approx O(e^{-k})$ that can be found in (39). To be more specific,$\|x_k - x^*\|_2 \leq sBe^{-ck}$. s is the sparsity level, B is maximum absolute value of the true signal, $c = -log(2\mu s - \mu)$ where μ is the mutual coherence of matrix A.

In the experiment, we worked with a rectangle measurement matrix $A \in \mathbb{R}^{m\times n}$. Matrix A is obtained by randomly sampling a Gaussian distribution where $A_{ij} \sim N(0, 1)$, $i, j \in \{1, ..., n\}$. $x \in \mathbb{R}^n$, where $n = 200, m = 100$, was selected as 2-sparse where nonzero elements were 1. After obtaining a training set of 4000 pairs of $\{x, y\}$ and a validation set of 50 pairs of $\{x, y\}$, 5 different networks are trained with different layer numbers:1,2,4,8,16. Then, results for $\|x^k - x_{true}\|_2$ from the validation set are obtained,averaged and compared with the theoretical bounded that is provided in (39). In theoretical bound, s is chosen as 2, B is



(a) (b) (c)

Figure 9: Convergence rate analysis for FISTA algorithm with measurements y with noise $N(0, 0.2)(\lambda = 1)$. (a) Reconstruction error, (b) Linear scale convergence rate and lower bound on the rate, (c) Log scale convergence rate and lower bound on the rate.



(a) (b)

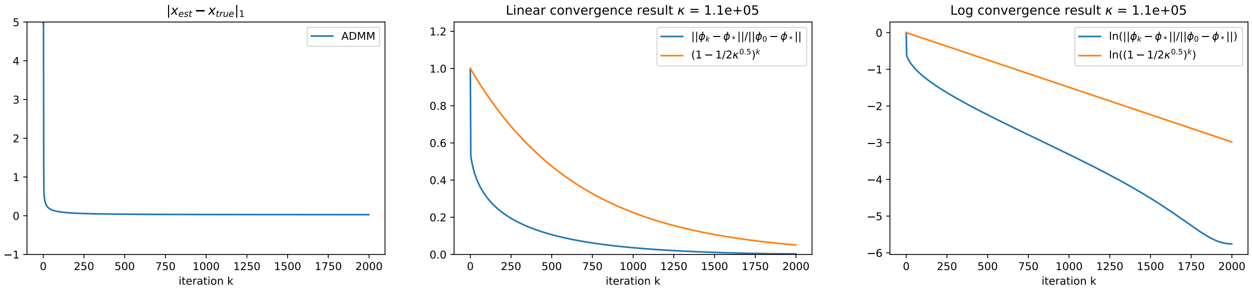
Figure 10: Convergence rate analysis for LISTA algorithm. (a) Noiseless setting, (b) Noisy setting with $N(0, 0.2)$

chosen 1. c that is the mutual coherence of matrix A is approximated computationally. In order to estimate mutual coherence, 2 Gaussian vectors are generated and normalized for 100000 times. Then largest inner product value is taken as the mutual coherence of matrix A that is 0.33. Respective lower bound on the rate was verified for measurements without noise that can be seen in Figure (10). One observation is that increasing layer number from 4 to 8 and 16, does not make a significant difference in empirical case. Our explanation for that is as the network size increases, the training requires additional effort due to overfitting. We can either increase training set size or add some other forms of regularization.

9.2.5 ADMM

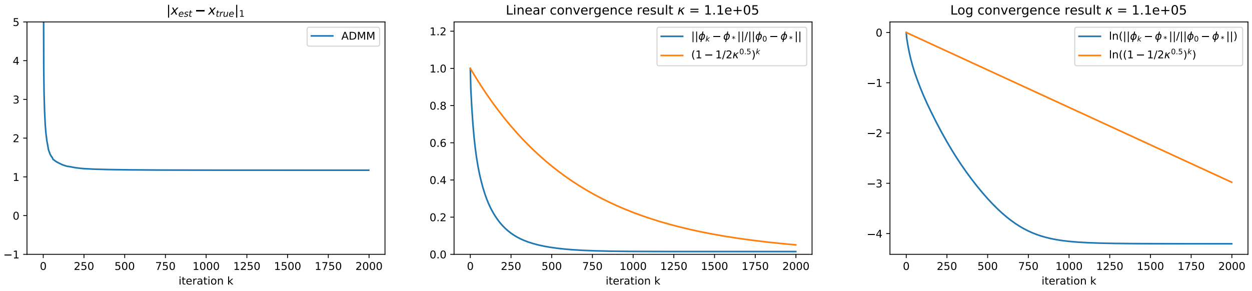
In section 6.1.3, we provided the theoretical convergence rate analysis for ADMM as a general lower bound where the method uses semidefinite programming. The derivation was for over-relaxed version of ADMM but by adjusting the parameter $\alpha = 1$ we were able to obtain the results for ADMM. Also, by adjusting other parameters of choice accordingly, we obtained the lower bound on the convergence rate for LASSO problem solved by ADMM as (54), where κ was the ratio of largest and smallest eigenvalues of measurement matrix $A^T A$.

In the experiment, we worked with a square measurement matrix $A \in \mathbb{R}^{n \times n}$ to obtain a finite condition number for $A^T A$ for analysis. A is obtained by randomly sampling a Gaussian distribution where $A_{ij} \sim N(0, 1)$, $i, j \in \{1, ..., n\}$. $x \in \mathbb{R}^n$, $n = 200$, was selected as 5-sparse where nonzero elements were 1, $\rho = \sqrt{\lambda_{A^T A_{max}} \lambda_{A^T A_{min}}}$ and $\kappa = \lambda_{A^T A_{max}} / \lambda_{A^T A_{min}}$ was taken in accordance with the derivation. Then, results for $||\varphi_k - \varphi_*|| / ||\varphi_0 - \varphi_*||$ are obtained and compared with the term $(1 - \frac{1}{2\kappa^{0.5}})^k$ where k is the iteration number and φ_* is the optimum value for the respective vector as indicated in Theorem 6.1. To obtain the fixed point value of dual variable, u_*, initially the algorithm runs once until complete convergence and optimum value of u after the algorithm converges is used as u_*. Respective lower bound on the rate was verified for measurements with and without noise for two cases that can be seen in Figure (11) and (12).



(a) (b) (c)

Figure 11: Convergence rate analysis for ADMM algorithm with measurements y without noise. Reconstruction error (a), linear convergence rate and lower bound on the rate (b), log convergence rate and lower bound on the rate (c).

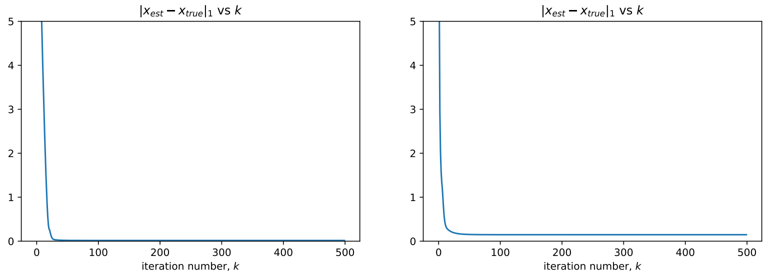


(a) (b) (c)

Figure 12: Convergence rate analysis for ADMM algorithm with measurements y with Gaussian noise $N(0, 0.2)$. Reconstruction error (a), linear convergence rate and lower bound on the rate (b), log convergence rate and lower bound on the rate (c).

9.2.6 PnP

In PnP, since denoiser model can be modified according to the task and preference, we can only analyze the fixed point convergence as we discussed in section 7.2. For this purpose, we analyzed the fixed point convergence property of the algorithm with and without noisy measurements, b, with FISTA algorithm implemented in denoising step, rather than soft thresholding. The selection of FISTA was justified by having another LASSO step while updating the split variable, z in the ADMM algorithm. In Figure 13, results can be observed for two cases where $x \in \mathbb{R}^n$ was 5-sparse and every element of the measurement matrix $A \in \mathbb{R}^{n \times n}$ was selected from normal distribution and was scaled such that its largest eigenvalue is 1 according to the assumption 3 in section (7.2).



(a) (b)

Figure 13: Fixed point convergence analysis for PnP algorithm with FISTA as denoiser step. (a) Measurements, b, without noise (b) with noisy measurements b.

Empirical results comply with the theory and for clean measurements, the algorithm converges to the exact x. For noisy measurements, algorithm does not exactly converge to x but it converges to a fixed point.

9.3 Empirical comparison of algorithms on the different problem settings

9.3.1 Problem 1: Sparse recovery

In this problem setting, we compared five different algorithms namely ISTA, FISTA,LISTA,ADMM and PnP on a fixed sparse recovery problem where $A \in R^{m\times n}$;$x \in R^n$; $b, n \in R^m$. A was obtained by randomly sampling a Gaussian distribution where $A_{ij} \sim N(0, 1)$. m was chosen as 100 and n was chosen 200.

$$b = Ax + n \tag{64}$$

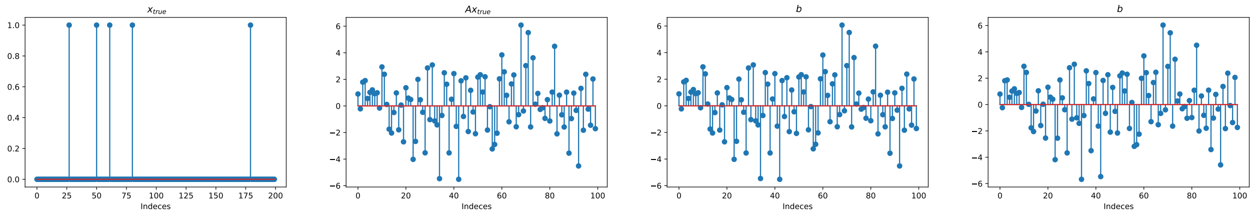
Two different scenarios were investigated: (i) low noise level $N(0, 0.01)$ and (ii) high noise level $N(0, 0.1)$. For both cases, non-zero elements of true signal x were 1 and sparsity level was chosen as 5. True signal and regarding measurement signals can be found in Fig (14).

For low level noise case, four different algorithms results are summarized in Figure 15. All the algorithms recover well under low noise case. For high level noise case, four different algorithms results are summarized in Figure 16. Our observation is that ISTA, FISTA and ADMM converges to similar recovery error, on the other hand, PnP converges to a worse recovery error.

In Figure 17, empirical convergence rates can be found. For both cases, PnP has the fastest convergence rate. Then, FISTA and ADMM have slightly slower convergence rate. Finally, ISTA has the slowest convergence rate.

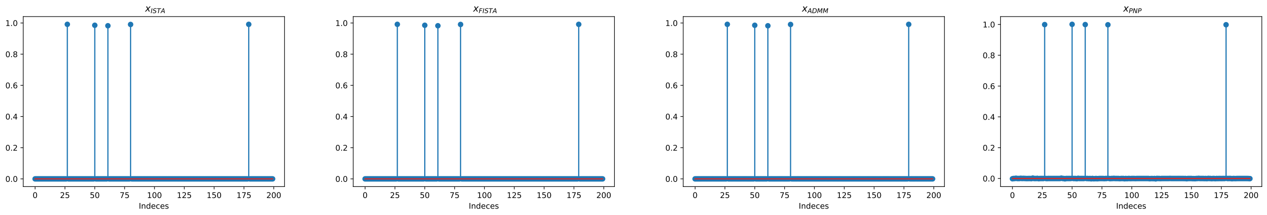
We have calculated the empirical required number iterations to reach ϵ approximate solutions. E.g. $\epsilon = 0.2$, then PnP converges in 17 steps, ADMM converges in 69 steps, FISTA converges in 87 steps and ISTA converges in 475 steps. For $\epsilon = 0.1$, then PnP converges in 92 steps, ADMM converges in 273 steps, FISTA converges in 92 steps and ISTA converges in 503 steps.

In Figure 18 and 19, recovery results of LISTA for different number of layers can be seen. In Figure 18, LISTA network is trained with noisy data where noise is sampled from $N(0, 0.01)$. In Figure 19, LISTA network is trained with noisy data where noise is sampled from $N(0, 0.1)$. For each case, three different networks are trained with different number of layers: (i)1, (ii)2 and (iii)4. As number of layer increases, recovery gets better.



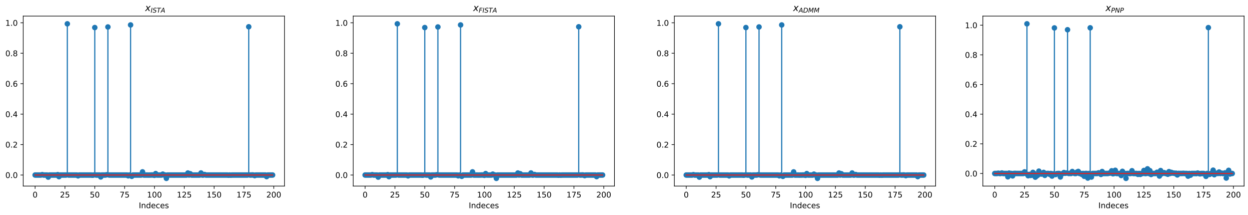
(a) (b) (c) (d)

Figure 14: True signal and measurements with two different noise levels. (a) x, (b) Ax, (c) b where $n \sim N(0, 10^{-2})$, (d) b where $n \sim N(0, 10^{-1})$.



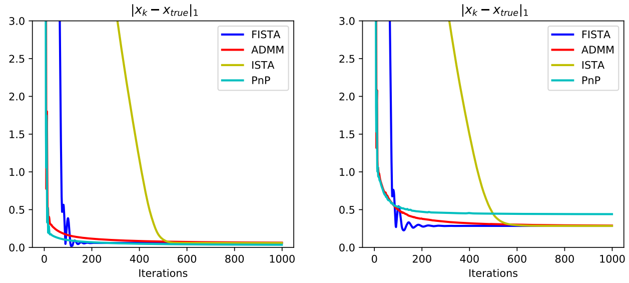
(a) (b) (c) (d)

Figure 15: Recovered signals using different algorithms via LASSO for $n \sim N(0, 10^{-2})$. (a) x_{ISTA}, (b) x_{FISTA}, (c) x_{ADMM} (d) $x_{PnP-FISTA}$.



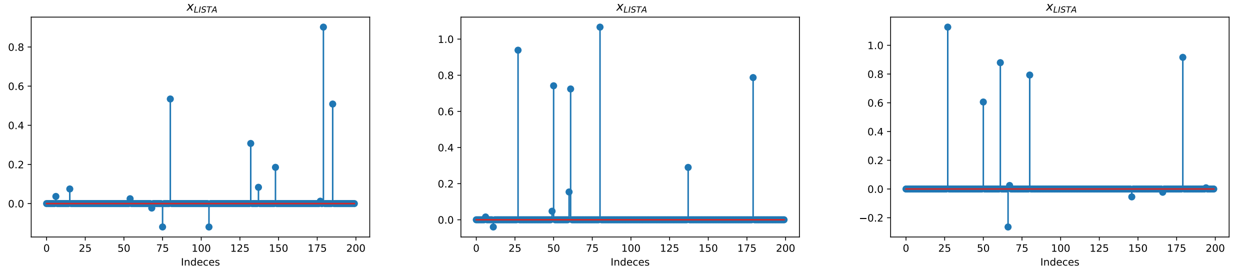
(a) (b) (c) (d)

Figure 16: Recovered signals using different algorithms via LASSO for $n \sim N(0, 10^{-1})$. (a) x_{ISTA}, (b) x_{FISTA}, (c) x_{ADMM} (d) $x_{PnP-FISTA}$.



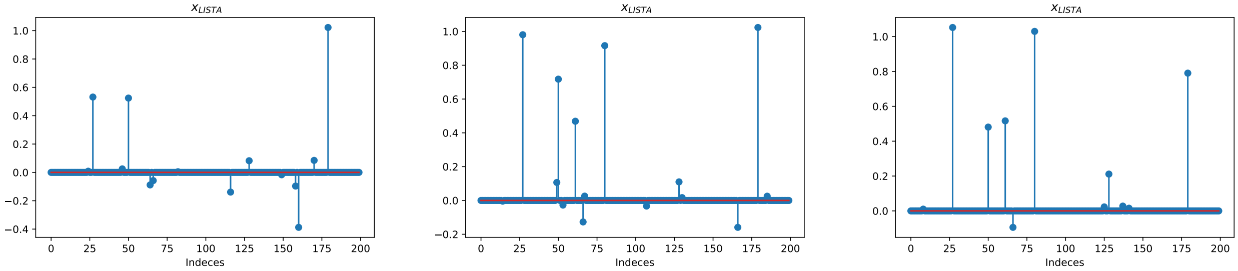
(a) (b)

Figure 17: Empirical Convergence Rates (a) $N(0, 0.01)$, (b) $N(0, 0.1)$



(a) (b) (c)

Figure 18: LISTA Recovery for Different Layer Numbers. (a) number of layers = 1 , (b) number of layers = 2 , (c) number of layers = 4



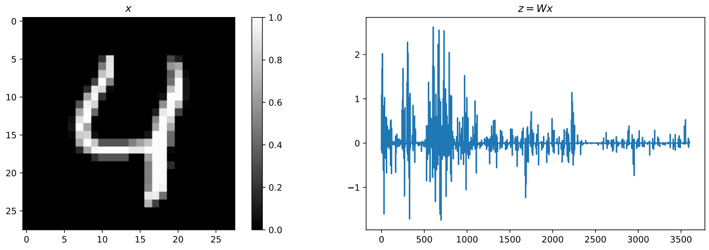
(a) (b) (c)

Figure 19: LISTA Recovery for Different Layer Numbers. (a) number of layers = 1 , (b) number of layers = 2 , (c) number of layers = 4

9.3.2 Problem 2: LASSO with wavelet decomposition prior

After modifying the objective function as in (13), we compared the performances of the respective algorithms on this problem by taking $F = W$, where W is a DWT matrix. Since size of the image puts a constraint on the memory and computation time, we decided to work with a smaller sized image obtained from MNIST dataset. The image, x, is 28×28. The measurement matrix A for sparse reconstruction case has the size of $\frac{28^2}{2} \times 28^2$. For deblurring setting, $A \in \mathbb{R}^{28^2 \times 28^2}$ was used. Figure 20 demonstrates the sparse nature of wavelet transform for an image from MNIST dataset.

Notes: Convergence plots for this problem setting in sections 9.3.2 and 9.3.2 were plotted until the iteration where the error was assigned to its minimum level. Also, plots were obtained using a colorbar that has a range between (0,1) but errors due nonnegative estimations still contribute to the overall error whereas they do not show up in the figures.

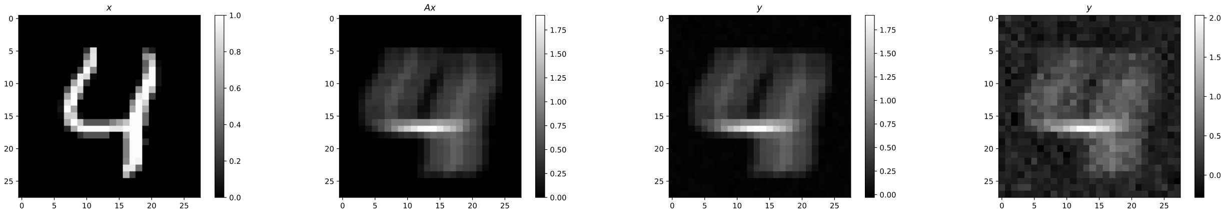


(a) (b)

Figure 20: Unsorted wavelet transform of an MNIST image. (a) x, (b) $z = Wx$ where W is DWT transform matrix.

Deblurring LASSO with wavelet decomposition prior:

In this problem setting, we generated a 3×3 normalized Gaussian kernel for horizontal blurring and converted it to Toeplitz form to obtain $A \in \mathbb{R}^{28^2 \times 28^2}$ and perform blurring on the 28×28 normalized image $x \in \mathbb{R}^{28 \times 28}$ obtained from MNIST dataset. Measurements, $b \in \mathbb{R}^{28 \times 28}$ are obtained using (64) a low level of noise and a high level of noise, where $n \sim N(0, 10^{-2})$ and $N(0, 5(10^{-1}))$ respectively.



(a) (b) (c) (d)

Figure 21: Original and blurred images with two different noise levels. (a) x, (b) Ax, (c) y where $n \sim N(0, 10^{-2})$, (d) y where $n \sim N(0, 5(10^{-2}))$.

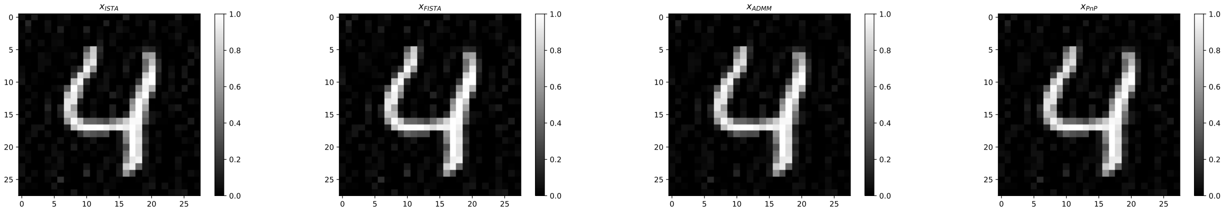
Then, $\lambda = 2(10^{-4})$ and $\lambda = 2(10^{-3})$ was selected using a similar approach as in section 9.2.1 for both noise levels respectively and problem was solved using ISTA, FISTA, ADMM and PnP-FISTA algorithms. Results are included in figures 22 and 24.

NOTE: $l2$ error plots were plotted until the point that the error decreased to its **minimum**.

When $n \sim N(0, 5(10^{-2}))$, λ needs to be increased in order to regularize the problem further and obtain meaningful results since the additive noise is substantially high. However, as we increase λ, $l1$-norm of the wavelet transform of the deblurred image gets suppressed and this behavior can be observed in the difference between results in figures 22 and 24.

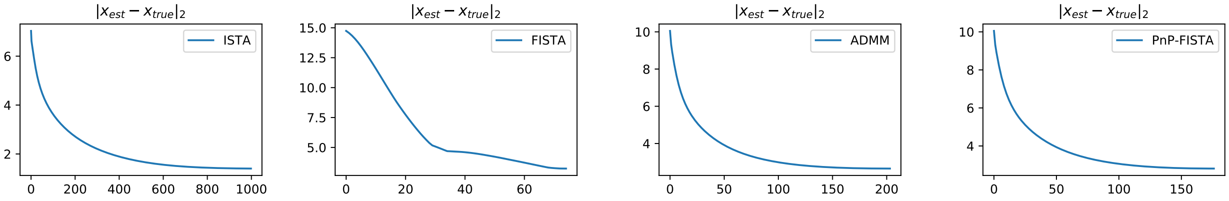
Sparse recovery via LASSO with wavelet decomposition prior:

In the second setting, we undersampled the same image $x \in \mathbb{R}^{28 \times 28}$ in figure 20 using a fat matrix $A \in \mathbb{R}^{\frac{28^2}{2} \times 28^2}$ with additive noise as in (64). Elements of A were sampled from $N(0, 10^{-1})$ and $\lambda = 5(10^{-2})$



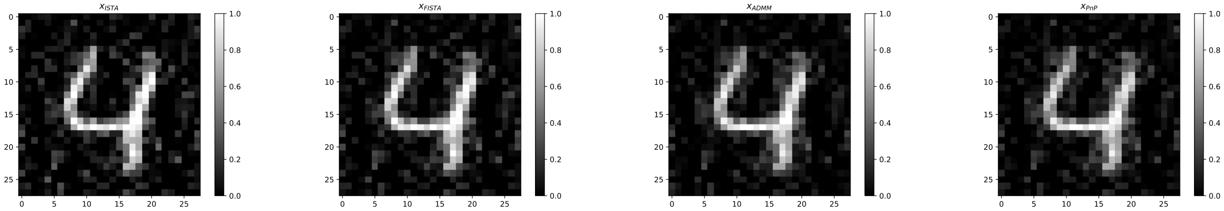
(a) (b) (c) (d)

Figure 22: Deblurred images using different algorithms via LASSO for $n \sim N(0, 10^{-2})$. (a) x_{ISTA}, (b) x_{FISTA}, (c) x_{ADMM} (d) $x_{PnP-FISTA}$.



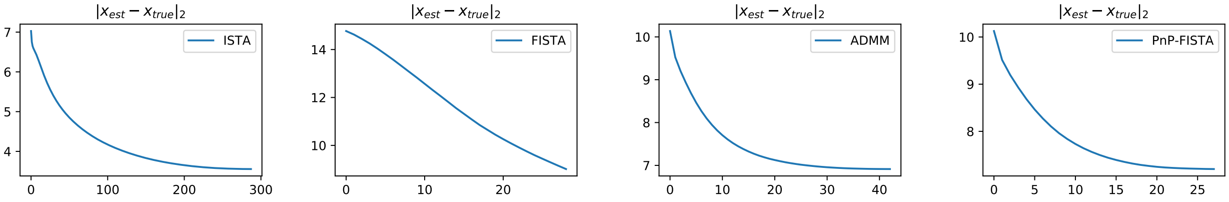
(a) (b) (c) (d)

Figure 23: $l2$ error vs iteration plots for different algorithms for deblurring via LASSO for $n \sim N(0, 10^{-2})$ with wavelet decomposition prior. (a) ISTA, (b) FISTA, (c) ADMM, (d) PnP-FISTA.



(a) (b) (c) (d)

Figure 24: Deblurred images using different algorithms via LASSO for $n \sim N(0, 5(10^{-2}))$. (a) x_{ISTA}, (b) x_{FISTA}, (c) x_{ADMM} (d) $x_{PnP-FISTA}$.



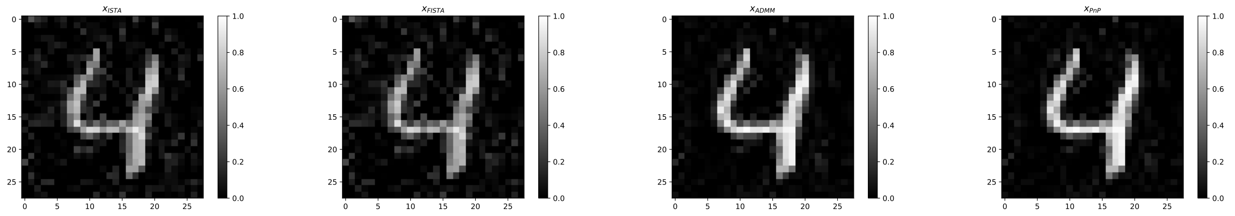
(a) (b) (c) (d)

Figure 25: $l2$ error vs iteration plots for different algorithms for deblurring via LASSO for $n \sim N(0, 5(10^{-2}))$ with wavelet decomposition prior. (a) ISTA, (b) FISTA, (c) ADMM, (d) PnP-FISTA.

was selected. Since the problem is already ill-posed, we did not play with the noise level for this case and proceeded with a relatively low noise of $n \sim N(0, 10^{-2})$.

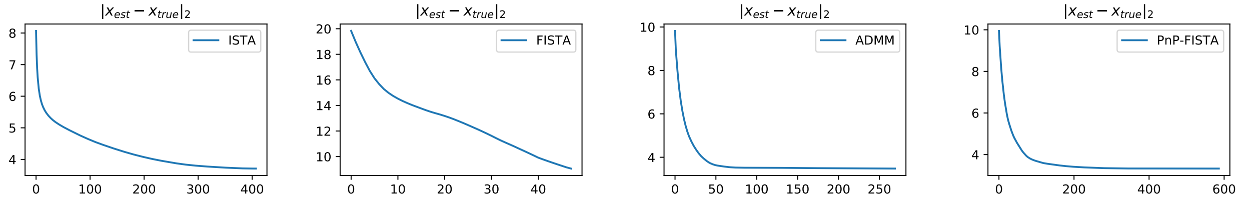
10 Discussion

In this section, we will have an overview of five selected algorithms which were compared under two different problem settings: (i) sparse recovery and (ii) deblurring by wavelet decomposition. By doing so, we will



(a) (b) (c) (d)

Figure 26: Reconstructed images using different algorithms via LASSO for $n \sim N(0, 10^{-2})$. (a) x_{ISTA}, (b) x_{FISTA}, (c) x_{ADMM} (d) $x_{PnP-FISTA}$.



(a) (b) (c) (d)

Figure 27: $l2$ error vs iteration plots for different algorithms for sparse recovery via LASSO for $n \sim N(0, 10^{-1})$ with wavelet decomposition prior. (a) ISTA, (b) FISTA, (c) ADMM, (d) PnP-FISTA.

reveal algorithms strengths and weaknesses for different applications.

In the first problem, we have implemented ISTA, FISTA, LISTA, ADMM and PnP for sparse recovery. In that problem, true signal is sparse and we are trying to solve an under-determined system for different noise levels. When we compare the algorithms in terms of computational cost per iterations, we can conclude that ADMM and PnP have the largest computational cost per iteration. ISTA, FISTA, LISTA have roughly the same computational cost per iteration. However, when we compare the algorithms in terms of number of iterations required to reach an ϵ approximate solution, we can conclude that LISTA has the fastest convergence rate which is linear. One caveat is that LISTA utilizes a specific training set and requires conscientious training to bring out its potential. Generalization can be a huge problem if one does not have a diverse training set. One observation regarding LISTA for this problem setting is that as we increase the layer number, the network does not generalize well. It is expected since the complexity of the network increases while degrees of freedom of the input data stays constant and therefore overfitting occurs. In order to overcome this problem, we have implemented early stopping and increased the training set size. Additionally, in this problem setting, we have observed that FISTA is more stable than ADMM and PnP. In other words, when we increase the noise level, recovery error of FISTA converges to a better point in a faster manner than others. Especially, final recovery error of PnP significantly increases as we increase the noise level, which is also expected because of the possible suboptimality of the denoiser implementation. However, fixed point convergence is observed in both cases which is consistent with the theoretical analysis of the algorithm. Overall, in this problem setting, we would prefer to apply FISTA when we consider its convergence rate, implementation simplicity, stability and recovery success.

In the second problem setting, we implemented ISTA, FISTA, ADMM, and PnP with FISTA denoiser step, excluding LISTA because of the scarcity of input data diversity. In the problem we had an image from MNIST dataset and a forward model A where the input-output relationship is defined as in 64 and the prior information related to such images being sparse wavelet transforms was used to solve problem with modified LASSO formulation. This problem had two subsettings. In the first one, we selected the forward model as a Gaussian blur kernel which was applied horizontally to the input image. In the second setting, we tried to perform sparse recovery where the measurement matrix A was a fat matrix with i.i.d. elements. Computational cost requirement per iteration comparison of the respective algorithms is the same as in first problem.

After selecting the step sizes for ISTA and FISTA optimally in coherence with the theoretical analysis, we observed that for deblurring case, recovery error of FISTA was slightly more than the other algorithms. This can be explained by the iterations of FISTA with "momentum". Due to this effect, algorithm converges faster but it alternates around the minimum. As we increase the noise, the problem becomes more ill-posed and further regularization (by increasing λ) is required to obtain optimal results. PnP-FISTA converges slightly faster than ADMM, which means that FISTA as denoiser selection has improved the convergence rate for this problem, rather than using the proximal mapping and ISTA provides the smallest reconstruction error but converges with a suboptimal rate.

In the sparse recovery setting, we obtained a better reconstruction from PnP-FISTA algorithm but convergence rate decreased for it. Again, FISTA obtains the fastest convergence to the minimum error but obtained result is suboptimal. Since the wavelet transform of the image as shown in Figure 20 is not as sparse as the exactly 5-sparse signal that we used in first problem, reconstruction errors are generally suboptimal for this setting as expected. For this setting, ADMM or PnP-FISTA would be a good choice for eventual reconstruction quality but they still suffer from high computation time.

11 Conclusion

In this project, we attempted to investigate several different widely used methods to solve LASSO problem. Initially we examined LASSO problem shortly and provided information related to LASSO in the perspective of the experimental settings that we used in the project. Then, we provided detailed information related to the algorithms that we implemented and investigated throughout the project and provided lower bound derivations for their convergence rates and proofs of convergence when necessary. After that, we obtained computational costs for respective algorithms and conducted computational experiments for different purposes. These computational experiments covered experimental verification of measurement bound for random Gaussian matrix since we use such a matrix in the following parts, empirical analysis and verification of the convergence rates and rate lower bounds for the algorithms and empirical comparison of algorithms on two different problem settings.

References

[1] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.

[2] Ronald E Bruck Jr. On the weak convergence of an ergodic iteration for the solution of variational inequalities for monotone operators in hilbert space. *Journal of Mathematical Analysis and Applications*, 61(1):159–164, 1977.

[3] Stanley H Chan, Xiran Wang, and Omar A Elgendy. Plug-and-play admm for image restoration: Fixed-point convergence and applications. *IEEE Transactions on Computational Imaging*, 3(1):84–98, 2016.

[4] Xiaohan Chen, Jialin Liu, Zhangyang Wang, and Wotao Yin. Theoretical linear convergence of unfolded ista and its practical weights and thresholds. In *Advances in Neural Information Processing Systems*, pages 9061–9071, 2018.

[5] Mark A Davenport, Marco F Duarte, Yonina C Eldar, and Gitta Kutyniok. Introduction to compressed sensing. *Compressed sensing: theory and applications*, 105:106, 2012.

[6] Francisco Facchinei and Jong-Shi Pang. *Finite-dimensional variational inequalities and complementarity problems*. Springer Science & Business Media, 2007.

[7] Karol Gregor and Yann LeCun. Learning fast approximations of sparse coding. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, pages 399–406. Omnipress, 2010.

[8] Laurent Lessard, Benjamin Recht, and Andrew Packard. Analysis and design of optimization algorithms via integral quadratic constraints. *SIAM Journal on Optimization*, 26(1):57–95, 2016.

[9] Jean-Jacques Moreau. Proximité et dualité dans un espace hilbertien. *Bulletin de la Société mathématique de France*, 93:273–299, 1965.

[10] Robert Nishihara, Laurent Lessard, Benjamin Recht, Andrew Packard, and Michael I Jordan. A general analysis of the convergence of admm. *arXiv preprint arXiv:1502.02009*, 2015.

[11] Sangjun Park and Heung-No Lee. On the derivation of rip for random gaussian matrices and binary sparse signals. In *ICTC 2011*, pages 120–124. IEEE, 2011.

[12] Gregory B Passty. Ergodic convergence to a zero of the sum of monotone operators in hilbert space. *Journal of Mathematical Analysis and Applications*, 72(2):383–390, 1979.

[13] Singanallur V Venkatakrishnan, Charles A Bouman, and Brendt Wohlberg. Plug-and-play priors for model based reconstruction. In *2013 IEEE Global Conference on Signal and Information Processing*, pages 945–948. IEEE, 2013.