Paper Summary

Sparse Signal Estimation by Maximally Sparse Convex Optimization

Chinchuthakun Worameth (18B00033)

November 18, 2021

This report summarizes the main contents in the paper *Sparse Signal Estimation by Maximally Sparse Convex Optimization* by Selesnick and Bayram in 2014, published in *IEEE Transactions on Signal Processing* [1]. The content is organized as follows. After a brief comparison between existing approaches and this work in Section 1, the proposed algorithm is presented in Section 2. Then, experiments and evaluation methods are described in Section 3. Finally, the remaining challenges and paper recommendations are provided in Section 4.

1 Background

1.1 Existing Approaches

Many applications in sparse signal processing can be represented by a linear model (1a) and framed as an ill-posed linear inverse problem (1b).

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{w} \tag{1a}$$

$$\arg\min_{\mathbf{x}\in\mathbb{R}^N} \Big(F(\mathbf{x}) = \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 + \mathbf{Reg} \Big) \tag{1b}$$

where $\mathbf{x} \in \mathbb{R}^N$, $\mathbf{y} \in \mathbb{R}^M$, $\mathbf{H} \in \mathbb{R}^{M\times N}$, and $\mathbf{w} \in \mathbb{R}^N$ are the original sparse signal, the observed signal, a linear operator, and an additive white Gaussian noise (AWGN), respectively. Note that **Reg** denotes a penalty function enforcing sparse regularization.

While ℓ_1 regularization, defined in (2) for $\lambda > 0$, is commonly employed to promote sparsity because of its simplicity and convexity, many studies have proposed alternatives that induce sparsity more strongly; thus, yielding better performance.

$$\arg\min_{\mathbf{x}\in\mathbb{R}^N} \Big(F(\mathbf{x}) = \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 + \lambda\|\mathbf{x}\|_1 \Big) \tag{2}$$

At the time this paper was written, there are two main research directions, as follows.

- **Non-differentiable convex penalty**

  As discussed throughout the course *Signal Processing and Optimization (ART.T465)*, many algorithms have been developed to efficiently solve (1b). Some examples are proximal gradient method, alternating direction method of multipliers (ADMM), and Primal-dual proximal splitting method.

- **Non-convex penalty**

  Researches in this direction focus on non-convex penalty functions that encourage sparsity more strongly than the ℓ_1 norm and non-convex optimization algorithms for solving them, i.e. avoiding local optima in the loss landscape as much as possible. Some examples are iterative reweighted least-squares (IRLS), FOCUSS, and graduated non-convexity (GNC).

1.2 Research Objective

Designing a good sparsity-promoting penalty function can be quite challenging. Convex penalties guarantee the global optimum but enforce looser sparsity constraints. Conversely, non-convex alternatives introduce local optima and are sensitive to the values of observed signal $\mathbf{y}$. To avoid this dilemma, the authors aim to introduce a penalty that induces sparsity more strongly than the ℓ_1 norm while preserving the convexity of the cost function. Mathematically, they consider the optimization problem

$$\arg\min_{\mathbf{x}\in\mathbb{R}^N} \left(F(\mathbf{x}) = \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 + \sum_{n=0}^{N-1} \lambda_n \phi_n(x_n; a_n) \right) \tag{3}$$

where $\lambda_n > 0$ and $\phi_n : \mathbb{R} \to \mathbb{R}$ for $n \in \mathbb{Z}_n$ and select parametric non-convex penalty functions $\phi_n(\cdot; a_n)$ based on the choice of $\mathbf{x}$ to ensure that F is convex.

According to the authors, the proposed technique can be regarded as the *generalization of GNC to non-smooth penalties*. Specifically, there are two distinctions: (1) the previous work considers penalty functions that are quadratic around the origin while this work extends to those that are non-differentiable at the origin, (2) this work removes the constraint on a_n, i.e. $a_i = a_j, \forall i, j \in \mathbb{Z}_N$. The novelty of this article lies in the application of the semidefinite program (SDP) for choosing ϕ_n. It provides a framework for systematically enhancing the sparsity-inducing capability of ℓ_1 norm while ensuring that it can be efficiently solved by leveraging its convexity. In other words, the worst-case performance of the proposed algorithm is equivalent to that of ℓ_1 norm regularization.

2 Methodology

2.1 Penalties and Threshold Functions

2.1.1 Problem Statement

For a given penalty function ϕ, its proximal operator, i.e. threshold function, $\theta : \mathbb{R} \to \mathbb{R}$ is defined as

$$\theta(y) = \arg\min_{x\in\mathbb{R}} \left(F(x) = \frac{1}{2}(y - x)^2 + \lambda\phi(x) \right) \tag{4}$$

where $\lambda > 0$. Following assumptions are made: (1) F is strictly convex, (2) $\phi(x)$ is three times continuously differentiable for all $x \in \mathbb{R} - \{0\}$, and (3) $\phi(-x) = \phi(x)$.

The authors are interested in a threshold function and its corresponding penalty that: (1) have a controllable value of $\theta'(T^+) = \max_{y\in\mathbb{R}} \theta'(y)$ from zero to infinity, and (2) $\lim_{y\to\infty} \big(y - \theta(y)\big) = 0$. The former mitigate spurious noise peaks which are often generated by hard-threshold functions with high $\theta'(T^+)$. The latter ensures that the penalty does not excessively attenuate peaks of the original signal. Note that the second property limits the choice of ϕ to only non-convex functions according to the following.

Proposition 1 *Suppose $\phi : \mathbb{R} \to \mathbb{R}$ is a convex function and $\phi(y)$ denotes the proximal operator associated with ϕ, defined in* (4)*. If $0 \le y_1 \le y_2$, then $y_1 - \theta(y_1) \le y_2 - \theta(y_2)$.*

2.1.2 Properties

The proximal operator ϕ defined in (4) can be expressed as

$$\theta(y) = \begin{cases} 0, & |y| < Y \\ f^{-1}(y), & |y| \ge Y \end{cases} \tag{5}$$

where $T = \lambda\phi'(x)$ is the threshold and $f : \mathbb{R}_+ \to \mathbb{R}$, defined as $f(x) = x + \lambda\phi'(x)$. The strictly convexity implies

$$\phi''(x) > -\frac{1}{\lambda}, \forall x > 0 \tag{6}$$

Moreover, the following are derived:

$$\theta'(T^+) = \frac{1}{1+\lambda\phi''(0^+)} \tag{7a}$$

$$\theta''(T^+) = \frac{\lambda\phi'''(0^+)}{(1+\lambda\phi''(0^+))^3} \tag{7b}$$

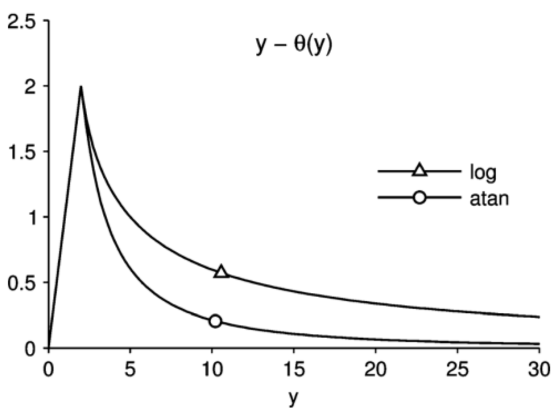
2.1.3 Logarithmic and Arctangent Penalties

The study focuses on two specific penalty functions, logarithmic and arctangent, which are described in (8) and (9), respectively. Note that $\lambda > 0$ and $0 < a \le 1/\lambda$.

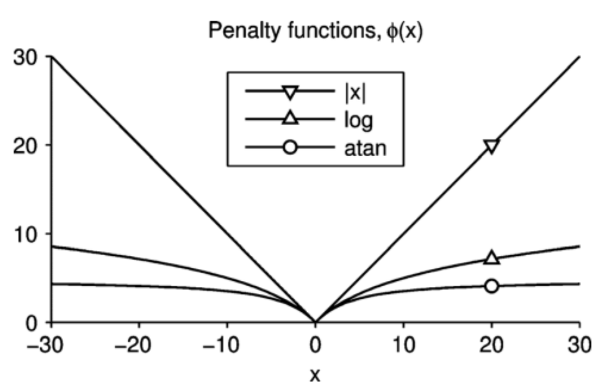
$$\phi(x) = \frac{1}{a}\log(1 + a|x|) \tag{8}$$

$$\phi(x) = \frac{2}{a\sqrt{3}}\left(\tan^{-1}\left(\frac{1+2a|x|}{\sqrt{3}} - \frac{\pi}{6}\right)\right) \tag{9}$$

Figure 1a and 1b show comparisons between logarithmic and arctangent penalty functions. The gap between identity function and the logarithmic penalty function approaches zero slower than arctangent. Moreover, the logarithmic tend to $+\infty$ while arctangent converges to a constant. For these reasons, arctangent is more suitable as a penalty function.



(a) Distance to identity function, $\theta'(T^+) = 2$



(b) Visualization of different penalties ($a = 0.25$)

Figure 1: Comparison of logarithmic penalty and arctangent penalty

2.2 Sparsity Penalized Least Squares

2.2.1 Convexity Condition

For a penalty function ϕ, define *parameter set associated with* ϕ, $\mathcal{S} = \{(\lambda, a) | v(x) = \frac{1}{2}x^2 + \lambda\phi(x; a) \text{ is strictly convex}\}$. Based on this definition, the condition ensuring that the function F described in (3) is strictly convex is

Proposition 2 *Suppose* $\mathbf{R}$ *is a positive definite diagonal matrix such that* $\mathbf{H}^T\mathbf{H} - \mathbf{R}$ *is positive semidefinite. Let* r_n *denote the* n*-th diagonal entry of* $\mathbf{R}$*, i.e.,* $[\mathbf{R}]_{n,n} = r_n > 0$*. Also, let* $\mathcal{S}$ *be the parameter set associated with phi. If* $(\lambda_n/r_n, a_n) \in \mathcal{S}$ *for each* n*, then* F *in* (3) *is strictly convex.*

For the proposed logarithmic and arctangent penalties, (8) and (9), the constrain becomes

$$0 < a_n < \frac{r_n}{\lambda_n} \tag{10}$$

When $\mathbf{H}$ does not have full rank, r_n will become zero for some indices n. In such cases, the authors define $\phi(x;0) = |x|$ to ensure convexity. This definition is justified for (8) and (9) because $\lim_{a\to 0}\phi(x;a) = |x|$. Hence, the problem turns into how to find appropriate values of $\mathbf{R}$ and λ_n for all indices n.

2.2.2 Calculation of R

One approach to calculate R is to formulate it as a convex optimization problem

$$\arg\max_{\mathbf{r}\in\mathbb{R}^N} \sum_{n=0}^{N-1} r_n \text{ s.t. } r_n \geq \alpha_{\min}, \mathbf{H}^T\mathbf{H} - \mathbf{R} \geq 0 \tag{11}$$

where $\alpha_{\min}$ is the minimum eigenvalue of $\mathbf{H}^T\mathbf{H}$, which incidentally proves that this problem has at least one solution because $\mathbf{R} = \alpha_{\min}\mathbf{I}$ satisfies both constraints. Since this type of convex optimization problems have been extensively studied, the authors use the MATLAB software package *SeDuMi* in their implementations.

2.2.3 Guideline for selecting λ

If F defined in (3) is strictly convex and ϕ is differentiable everywhere except zero, its global optimum $\mathbf{x}^*$ satisfies the condition

$$\begin{cases} \frac{1}{\lambda_n}\left[\mathbf{H}^T(\mathbf{y} - \mathbf{H}\mathbf{x}^*)\right]_n = \phi'(x_n^*; a_n), & x_n^* \neq 0 \\ \frac{1}{\lambda_n}\left[\mathbf{H}^T(\mathbf{y} - \mathbf{H}\mathbf{x}^*)\right]_n \in [\phi'(0^-; a_n), \phi'(0^+; a_n)], & x_n^* = 0 \end{cases} \tag{12}$$

where $[\mathbf{v}]_n$ denotes the n-th component of the vector $\mathbf{v}$.

When $\mathbf{x} = 0$ in (1a), it follows that $\mathbf{x}^* = 0$ in (3). In this case, (12) suggests choosing λ_n for ℓ_1, logarithmic, and arctangent penalty functions such that $|[\mathbf{H}^T\mathbf{w}]_n| \leq \lambda_n$. Therefore, the guideline is to set λ_n to the smallest possible values to avoid attenuation of x_n, which is

$$\lambda_n \approx \max|[\mathbf{H}^T\mathbf{w}]_n| \tag{13}$$

Although the actual value of $\mathbf{w}$ is unknown, it is justified to apply *three-sigma rule*, i.e. $\lambda_n \approx 3\mathbf{std}\big(|[\mathbf{H}^T\mathbf{w}]_n|\big)$. If $\mathbf{w} \sim \mathcal{N}(\cdot, \sigma^2)$, it follows that $\mathbf{std}\big(|[\mathbf{H}^T\mathbf{w}]_n|\big) = \sigma\|\mathbf{H}(\cdot, n)\|_2$ where $\mathbf{H}(\cdot, n)$ denotes n-th column of $\mathbf{H}$.

2.3 Maximally Sparse Convex Optimization and Its Extension

Following derivations above, the algorithm for maximally sparse convex optimization (MSC) is given. In step 2, although β can be any real numbers from 0 to 1 as indicated in (10), the paper recommends $\beta = 1$ because ϕ_n becomes maximally non-convex (maximally sparsity-inducing) while F is convex.

Algorithm 1: Maximally Sparse Convex Optimization (MSC)

Input : $\mathbf{y} \in \mathbb{R}^M, \mathbf{H} \in \mathbb{R}^{M\times N}$
Output : $\mathbf{x} \in \mathbb{R}^N$
Parameter: $\{\lambda_n > 0, n \in \mathbb{Z}_n\}, \phi : \mathbb{R}\times\mathbb{R} \to \mathbb{R}, \beta \in [0, 1]$
1. Solve (11) for $\mathbf{R}$ or use the sub-optimal solution $\mathbf{R} = \alpha_{\min}\mathbf{I}$.
2. Set $a_n = \beta r_n/\lambda_n, \forall n \in \mathbb{Z}_n$.
3. Solve (3) for $\mathbf{x}$.

One might notice that $\mathbf{R} \approx \mathbf{0}$ yields $\phi_n(x, a_n) = |x|$ for most indices n. In such cases, MSC does not offer any significant advantage over approaches based on ℓ_1 norm. To address this problem, the authors propose Iterative MSC (IMSC), described below.

Algorithm 2: Iterative Maximally Sparse Convex Optimization (IMSC)

Input : $\mathbf{y} \in \mathbb{R}^M, \mathbf{H} \in \mathbb{R}^{M \times N}$
Output : $\mathbf{x} \in \mathbb{R}^N$
Parameter: $\{\lambda_n > 0, n \in \mathbb{Z}_n\}, \phi : \mathbb{R} \times \mathbb{R} \to \mathbb{R}, \beta \in [0, 1], \mathit{iteration} \geq 1$
1. Initialize $\mathcal{K}^{(0)} = \{\}$.
2. Let $\mathbf{x}^{(1)}$ be the solution of (3) where $\phi_n(x, a_n) = |x|, \forall n \in \mathbb{Z}_n$.
for $i \leftarrow 1$ **to** *iteration* **do**
  3. Set $\mathcal{K}^{(i)} = \{n \in \mathbb{Z}_N | x_n^{(i)} \neq 0\}$.
  if $|\mathcal{K}^{(i)}| < |\mathcal{K}^{(i-1)}|$ **then break**;
  4. Let $\mathbf{H}^{(i)} \in M \times |\mathcal{K}^{(i)}|$ be the sub-matrix of $\mathbf{H}$ containing only columns $k \in \mathcal{K}^{(i)}$.
  5. Solve (11) where $\mathbf{H} = \mathbf{H}^{(i)}$ for $\mathbf{R}^{(i)} \in |\mathcal{K}^{(i)}| \times |\mathcal{K}^{(i)}|$ or use $\mathbf{R}^{(i)} = \alpha_{\min}^{(i)} \mathbf{I}$.
  6. Set $a_n^{(i)} = \beta r_n^{(i)} / \lambda_n, \forall n \in \mathcal{K}^{(i)}$.
  7. Solve (3) for the $|\mathcal{K}^{(i)}|$-dimensional convex problem to obtain $\mathbf{u}^{(i)}$.
  8. Set $x_n^{(i+1)} = \begin{cases} 0, & n \notin \mathcal{K}^{(i)} \\ u_n^{(i+1)}, & n \in \mathcal{K}^{(i)} \end{cases}$
end

---

3 Experiment

This paper includes two experiments on denoising and deconvolution. The former is not a notable experiment but rather a simple demonstration to illustrate the trade-off between $\theta'(T^+)$ and attenuation at peaks. The latter compares the performance of IMSC with other algorithms.

3.1 Denoising Experiment

Wavelet domain thresholding is performed on a part of the signal from WaveLab with length 2048. The noise is AWGN with $\sigma = 0.4$. Each threshold function uses the same threshold $T = 3\sigma$.

As shown in Figure 2a, the hard threshold (high $\theta'(T^+)$) yields the lowest root mean square error (RSME) with some undesirable spurious noise peaks. On the other hand, soft thresholding attenuates the peaks and achieves a higher RSME but effectively suppresses the spurious noise peaks. Avoiding both ends of the spectrum, arctangent threshold function, $\theta(y)$ associated with (9), results in a lower RSME while slightly reducing noise peaks.

3.2 Deconvolution Experiment

A sparse signal $\mathbf{x} \in \mathbb{R}^{1000}$ is generated with the following constrains: (1) the inter-spike interval is an integer in range (5,35) and (2) the amplitude of each spike is uniform between -1 to 1. The observed data is generated by a linear time-invariant system

$$y_n = \sum_k b_k x_{n-k} - \sum_k a_k y_{n-k} + w_n \tag{14}$$

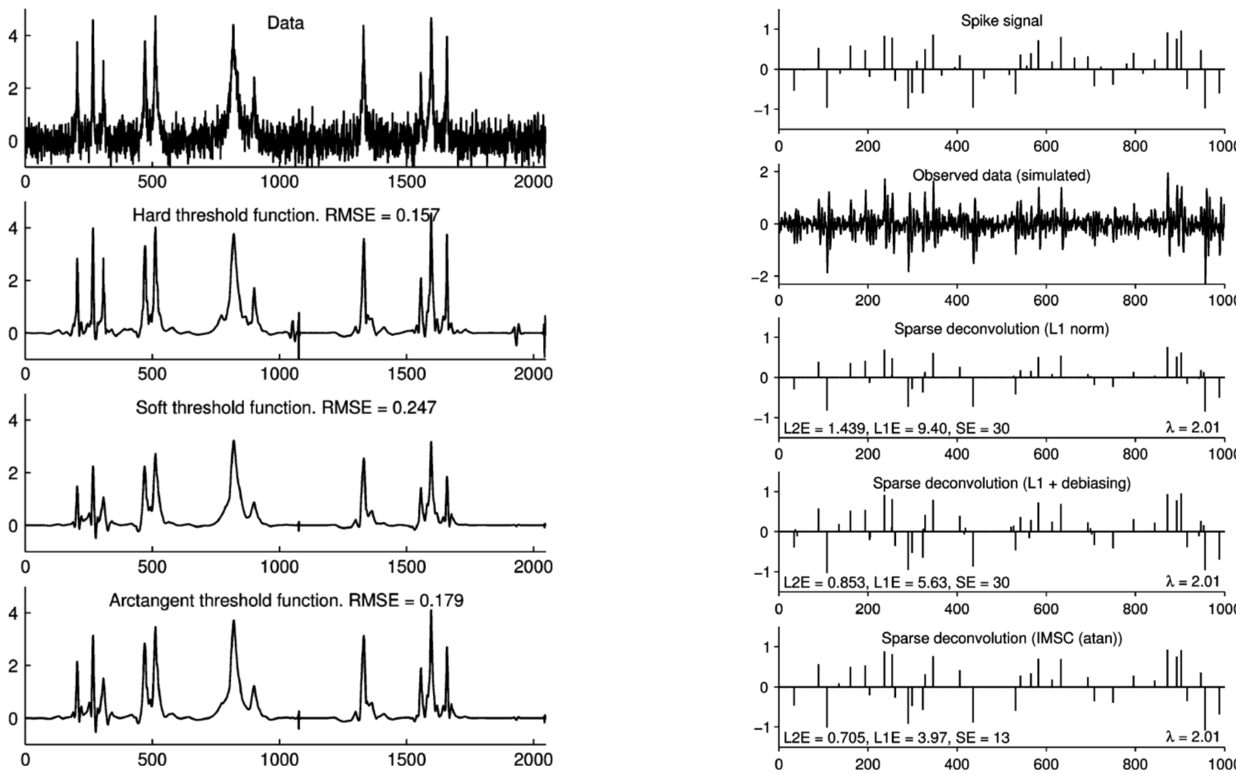
where $w_n \sim \mathcal{N}(0, \sigma^2)$ with $b_0 = 1, b_1 = 0.8, a_0 = 1, a_1 = -1.047, a_2 = 0.81$, and $\sigma = 0.2$.

Several algorithms listed in Figure 3a are evaluated with the following evaluation metrics: (1) l_2 norm (L2E), (2) ℓ_1 norm (L1E), (3) support error (SE), (4) number of false zeros in SE (FZ), and (5) number of false non-zeros in SE (FN). This experiment uses SE with $\epsilon = 10^{-3}$, which is defined by

$$\text{SE} = \|s(\mathbf{x}) - s(\hat{\mathbf{x}})\|_0 \tag{15a}$$

$$[s(\mathbf{x})]_n = \begin{cases} 1, & |x_n| > \epsilon \\ 0, & |x_n| \leq \epsilon \end{cases} \tag{15b}$$

where $\mathbf{x}$ and $\hat{\mathbf{x}}$ are the original and estimated signal, respectively. Visualization of $\mathbf{x}$, $\mathbf{y}$, and $\hat{\mathbf{x}}$ is shown in Figure 2b. Relationship between regularization parameters and evaluation metrics are investigated in Figure 3b. Note that debiasing refers to a post-processing step in which least squares is performed over the obtained support set.



(a) Denoising experiment

(b) Deconvolution experiment

Figure 2: Visualization of different signals

4 Conclusion

4.1 Future Works

Although MSC has proven to be an efficient sparse signal estimation algorithm, the authors mention some remaining challenges in the article.

1. **Incorporation with other techniques**

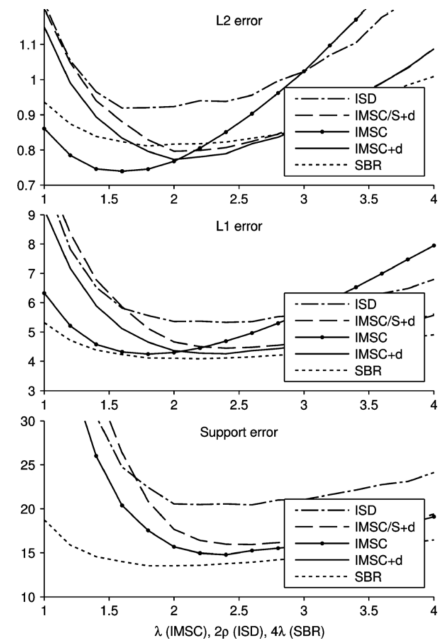
   Considering that iterative reweighted ℓ_1 norm outperforms its l_2 counterpart in Figure 3a, investigating reweighted MSC for non-convex optimization is a promising research direction.

2. **Application in large-scle problems**

   The current MSC implementation assumes that $\mathbf{H}$ is explicitly available. However, it is impractical for large-scale problems because it requires considerable computational resources to store and modify the matrix. Therefore, a novel algorithm is required for solving (3), i.e. calculating $\mathbf{H}^T\mathbf{H}$, without manipulating $\mathbf{H}$.

Algorithm	L2E	L1E	SE	(FZ, FN)
ℓ_1 norm	1.443	10.01	37.60	(10.3, 27.3)
ℓ_1 norm + debiasing	0.989	7.14	37.57	(10.3, 27.2)
AIHT [7]	1.073	6.37	24.90	(12.4, 12.5)
ISD [68]	0.911	5.19	19.67	(11.6, 8.1)
SBR [63]	0.788	4.05	13.62	(12.0, 1.6)
$\ell_p(p = 0.7)$ IRL2	0.993	5.80	16.32	(12.9, 3.4)
$\ell_p(p = 0.7)$ IRL2 + debiasing	0.924	4.82	16.32	(12.9, 3.4)
$\ell_p(p = 0.7)$ IRL1	0.884	5.29	14.43	(11.5, 2.9)
$\ell_p(p = 0.7)$ IRL1 + debiasing	0.774	4.18	14.43	(11.5, 2.9)
IMSC (log)	0.864	5.08	17.98	(9.8, 8.2)
IMSC (log) + debiasing	0.817	4.83	17.98	(9.8, 8.2)
IMSC (atan)	0.768	4.29	15.43	(10.0, 5.5)
IMSC (atan) + debiasing	0.769	4.35	15.42	(10.0, 5.5)
IMSC/S (atan)	0.910	5.45	17.93	(9.8, 8.1)
IMSC/S (atan) + debiasing	0.800	4.73	17.92	(9.8, 8.1)

(a) Average error (averaged over 200 trials)



(b) Error as functions of regularization parameter λ (averaged over 100 trials)

Figure 3: Deconvolutional experiment errors

4.2 Paper Recommendation

The proposed convexity-preserving non-convex penalties in this paper are separable (or additive). When $\mathbf{H}^T\mathbf{H}$ is singular, they provide only little improvement relative to the ℓ_1 norm [2]. Therefore, I strongly recommend interested readers to continue on another paper by the same leading author, *Sparse Regularization via Convex Analysis* [3]. It focuses on a bivariate non-separable non-convex penalty that preserves the convexity of the cost function.

References

[1] I. W. Selesnick and I. Bayram, "Sparse signal estimation by maximally sparse convex optimization," *IEEE Transactions on Signal Processing*, vol. 62, no. 5, pp. 1078–1092, 2014.

[2] ——, "Enhanced sparsity by non-separable regularization," *IEEE Transactions on Signal Processing*, vol. 64, no. 9, pp. 2298–2313, 2016.

[3] I. Selesnick, "Sparse regularization via convex analysis," *IEEE Transactions on Signal Processing*, vol. 65, no. 17, pp. 4481–4494, 2017.