L0Soft: ℓ_0 Minimization via Soft Thresholding

Mostafa Sadeghi*, Fateme Ghayem*, Massoud Babaie-Zadeh*, Saikat Chatterjee†, Mikael Skoglund†, Christian Jutten‡
*Electrical Engineering Department, Sharif University of Technology, Tehran, Iran
†Communication Theory Lab, KTH, Royal Institute of Technology, Stockholm, 10044, Sweden
‡GIPSA-lab, University of Grenoble, Grenoble, France

***Abstract*—We propose a new algorithm for finding sparse solution of a linear system of equations using ℓ_0 minimization. The proposed algorithm relies on approximating the non-smooth ℓ_0 (pseudo) norm with a differentiable function. Unlike other approaches, we utilize a particular definition of ℓ_0 norm which states that the ℓ_0 norm of a vector can be computed as the ℓ_1 norm of its sign vector. Then, using a smooth approximation of the sign function, the problem is converted to ℓ_1 minimization. This problem is solved via iterative proximal algorithms. Our simulations on both synthetic and real data demonstrate the promising performance of the proposed scheme.**

***Index Terms*—Compressed sensing, sparse representation, iterative hard thresholding, iterative soft thresholding, proximal algorithms**

I. Introduction

Finding sparse solutions of linear systems of equations has gained a lot of interest during the past decade [1]. This problem has appeared in a wide range of areas such as signal processing and computer vision, with applications ranging from signal enhancement and recovery [1] to pattern recognition and classification [2]. Compressed sensing [3], [4] and sparse signal representation [1] are two important applications of this problem. The sparsity seeking problem is usually formulated as

$$\min_{\mathbf{x}} \ \|\mathbf{x}\|_0 \quad \text{s.t.} \quad \|\mathbf{y}-\mathbf{A}\mathbf{x}\|_2 \le \epsilon \tag{1}$$

where, $\mathbf{y} \in \mathbb{R}^m$, $\mathbf{A} \in \mathbb{R}^{m\times n}$, $\epsilon \ge 0$, and $\|.\|_0$ denotes the ℓ_0 (pseudo) norm. Considering the fact that problem (1) is NP-hard [5], numerous alternative problems have been proposed [6], including ℓ_1 norm minimization [7], with various solvers yielding approximate solutions to (1); see [8]–[14]. For instance, consider the regularized version of (1), which is expressed as

$$\min_{\mathbf{x}} \ \frac{1}{2}\|\mathbf{y}-\mathbf{A}\mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_0, \tag{2}$$

with $\lambda > 0$. This problem can be attacked by proximal algorithms [15], as utilized by the popular iterative hard-thresholding (IHT) algorithm [16]. This amounts to performing the following iterative procedure

$$\mathbf{x}_{k+1} = H_{\lambda\cdot\mu}\big(\mathbf{x}_k - \mu\mathbf{A}^T(\mathbf{A}\mathbf{x}_k - \mathbf{y})\big), \tag{3}$$

This work has been supported by Iran National Science Foundation (INSF), no. 96000780, and by the European project ERC-2012AdG-320684-CHESS.

where, $\mu > 0$ is a step-size, and the entry-wise function $H_\lambda(.)$ denotes the hard thresholding operator [16], defined as

$$H_\lambda(x) \triangleq \begin{cases} x & |x| \ge \sqrt{2\lambda} \\ 0 & |x| < \sqrt{2\lambda} \end{cases}. \tag{4}$$

In this paper, we propose a new solver for (1), which is based on approximating the discontinuous ℓ_0 norm function with a differentiable one. Although there are several algorithms based on the same underlying idea, e.g., smoothed ℓ_0 (SL0) algorithm [17], here, we take a different path by utilizing a special definition of ℓ_0 norm based on the sign function, and approximating the sign function with a smooth one. This converts (1) into an ℓ_1 minimization problem which ends up with an iterative soft thresholding [1] algorithm. Simulation results on synthetic as well as real data demonstrate the promising performance of the proposed algorithm.

The rest of this paper is organized as follows. Section II presents the main idea and algorithmic description of the proposed solver. Then, Section III reports and discusses the results of our simulations.

II. Proposed method

A. Problem formulation

As our main motivation, note that the ℓ_0 function can be defined as

$$\|\mathbf{x}\|_0 = \sum_{i=1}^{n} |\text{sgn}(x_i)|, \tag{5}$$

where, "sgn" denotes the sign function and $\text{sgn}(0) \triangleq 0$. Equivalently, we can write

$$\|\mathbf{x}\|_0 = \|\mathbf{z}\|_1, \quad \mathbf{z} = \text{sgn}(\mathbf{x}), \tag{6}$$

with sgn acting entry-wise. Our main idea for solving (1) is then to make use of the above relation between $\mathbf{x}$ and $\mathbf{z}$. In this way, we arrive at the following equivalent form of (1)

$$\min_{\mathbf{x},\mathbf{z}} \ \|\mathbf{z}\|_1 \ \text{ s.t. } \begin{cases} \mathbf{z} = \text{sgn}(\mathbf{x}) \\ \|\mathbf{y}-\mathbf{A}\mathbf{x}\|_2 \le \epsilon \end{cases}. \tag{7}$$

The following subsection presents our proposed algorithm for solving (7)

B. Algorithm details

To solve (7), we consider the following regularized version of it, which is based on penalty methods [18]

$$\min_{\mathbf{x},\mathbf{z}} \|\mathbf{z}\|_1 + \frac{1}{2\alpha}\|\mathbf{z} - \operatorname{sgn}(\mathbf{x})\|_2^2 + \delta_{\mathcal{C}_\epsilon}(\mathbf{x}), \tag{8}$$

where, $\delta_{\mathcal{C}_\epsilon}(\mathbf{x})$ is the indicator function of $\mathcal{C}_\epsilon(\mathbf{x}) = \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2 \le \epsilon\}$ (taking a value of 0 when $\mathbf{x} \in \mathcal{C}_\epsilon$ and $+\infty$ otherwise), and $\alpha > 0$ is a penalty parameter. Note that problems (7) and (8) become equivalent when $\alpha \to 0$. Since the sign function is not differentiable, we use a smooth approximation of it. To this end, we consider the following hyperbolic tangent function:

$$f_\beta(x) \triangleq \tanh(\beta x) = \frac{\exp(2\beta x) - 1}{\exp(2\beta x) + 1}. \tag{9}$$

The sign function along with its smooth approximation for different values of β is plotted in Fig 1. As can be seen, larger values of β give tighter approximations to the sign function.

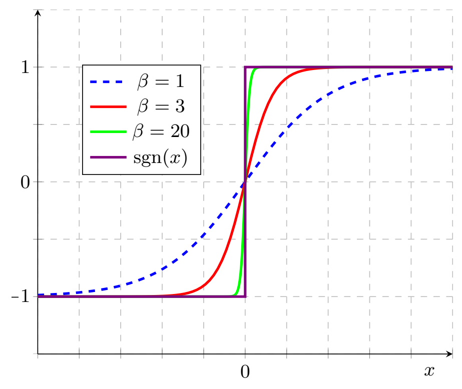

Fig. 1: Plots of the sign function and its smooth approximations, f_β defined in (9), for different values of β.

Therefore, in problem (8), we replace sgn with its smooth approximation. Doing so, we would have

$$\min_{\mathbf{x},\mathbf{z}} \alpha\|\mathbf{z}\|_1 + \frac{1}{2}\|\mathbf{z} - f_\beta(\mathbf{x})\|_2^2 + \delta_{\mathcal{C}_\epsilon}(\mathbf{x}), \tag{10}$$

in which, f_β acts entry-wise. Let us define

$$\begin{cases} F(\mathbf{x},\mathbf{z}) = \frac{1}{2}\|\mathbf{z} - f_\beta(\mathbf{x})\|_2^2 \\ g(\mathbf{x}) = \delta_{\mathcal{C}_\epsilon}(\mathbf{x}) \\ h(\mathbf{z}) = \alpha\|\mathbf{z}\|_1 \end{cases} \tag{11}$$

Then, our target problem (10) can be rewritten as

$$\min_{\mathbf{x},\mathbf{z}} \left\{ H(\mathbf{x},\mathbf{z}) \triangleq F(\mathbf{x},\mathbf{z}) + g(\mathbf{x}) + h(\mathbf{z}) \right\}. \tag{12}$$

In order to solve the above problem, we adopt a proximal alternating linearized minimization (PALM) approach [19]. This method assumes that the smooth part of the cost function is gradient Lipschitz with respect to each variable. That is, for problem (12), we should show that there exist some $L_x, L_z > 0$ such that $\forall \mathbf{x}, \mathbf{z}, \mathbf{u}, \mathbf{v}$:

$$\begin{cases} \|\nabla_x F(\mathbf{x},\mathbf{z}) - \nabla_x F(\mathbf{u},\mathbf{z})\|_2 \le L_x\|\mathbf{x} - \mathbf{u}\|_2 \\ \|\nabla_z F(\mathbf{x},\mathbf{z}) - \nabla_z F(\mathbf{x},\mathbf{v})\|_2 \le L_z\|\mathbf{z} - \mathbf{v}\|_2 \end{cases}. \tag{13}$$

To meet this objective, first note that

$$\begin{cases} \nabla_x F(\mathbf{x},\mathbf{z}) = f'_\beta(\mathbf{x}) \odot (f_\beta(\mathbf{x}) - \mathbf{z}) \\ \nabla_z F(\mathbf{x},\mathbf{z}) = \mathbf{z} - f_\beta(\mathbf{x}) \end{cases}. \tag{14}$$

Here, $\odot$ denotes Hadamard (entry-wise) product, and f'_β, the derivative of f_β, is given by

$$f'_\beta(x) = \beta \operatorname{sech}^2(\beta x), \quad \operatorname{sech}(x) = \frac{2}{\exp(x) + \exp(-x)}. \tag{15}$$

It is easy to verify that $L_z = 1$. To prove the existence of some L_x, consider the scalar function $J(x) = f'_\beta(x)\cdot(f_\beta(x) - z)$. We then derive an upperbound on the magnitude of its derivative

$$J'(x) = -2\beta^2 \cdot \operatorname{sech}^2(\beta x) \cdot \tanh(\beta x) \cdot (\tanh(\beta x) - z) + \beta^2 \cdot \operatorname{sech}^4(\beta x). \tag{16}$$

Noting that $\forall x$, $|\tanh(\beta x)| \le 1$, and using the identity $\operatorname{sech}^2(\beta x) = 1 - \tanh^2(\beta x)$, we arrive at $|J'(x)| \le \beta^2(3 + 2|z|)$. Finally, it is straightforward to show that

$$L_x = (3 + 2|z|) \cdot \beta^2 \tag{17}$$

satisfies (13).

Utilizing the PALM technique, the proposed algorithm starts with some $(\mathbf{x}_0, \mathbf{z}_0)$ and then generates a sequence $\{(\mathbf{x}_k, \mathbf{z}_k)\}_{k\ge1}$ to update $\mathbf{x}$ and $\mathbf{z}$. The details are given in the following subsections.

1) Updating $\mathbf{z}$*:* The update problem for $\mathbf{z}$ is

$$\mathbf{z}_{k+1} = \operatorname*{argmin}_{\mathbf{z}} \langle \nabla_z F(\mathbf{x}_k,\mathbf{z}_k), \mathbf{z} - \mathbf{z}_k\rangle + \frac{1}{2\mu_z}\|\mathbf{z} - \mathbf{z}_k\|_2^2 + h(\mathbf{z}), \tag{18}$$

where, $\langle \ , \ \rangle$ denotes inner-product, and $\mu_z \in (0, 1/L_z]$. Problem (18) can be simplified to

$$\mathbf{z}_{k+1} = \operatorname*{argmin}_{\mathbf{z}} \frac{1}{2}\|\mathbf{z} - \tilde{\mathbf{z}}_k\|_2^2 + \mu_z \cdot h(\mathbf{z}) \tag{19}$$

where, $\tilde{\mathbf{z}}_k = \mathbf{z}_k - \mu_z \nabla_z F(\mathbf{x}_k,\mathbf{z}_k) = (1 - \mu_z)\mathbf{z}_k + \mu_z f_\beta(\mathbf{x}_k)$. The optimal solution of (19) is characterized via the *soft-thresholding operator* [1] defined as

$$S_\lambda(x) \triangleq \begin{cases} x - \lambda & x > \lambda \\ 0 & |x| \le \lambda \\ x + \lambda & x < -\lambda \end{cases}. \tag{20}$$

The final update formula for $\mathbf{z}$ would then be

$$\mathbf{z}_{k+1} = S_{\mu_z\cdot\alpha}\big((1 - \mu_z)\cdot\mathbf{z}_k + \mu_z \cdot f_\beta(\mathbf{x}_k)\big). \tag{21}$$

2) Updating $\mathbf{x}$*:* To update $\mathbf{x}$, the following problem should be solved

$$\mathbf{x}_{k+1} = \underset{\mathbf{x}}{\operatorname{argmin}} \langle \nabla_x F(\mathbf{x}_k, \mathbf{z}_{k+1}), \mathbf{x} - \mathbf{x}_k \rangle + \frac{1}{2\mu_x} \|\mathbf{x} - \hat{\mathbf{x}}_k\|_2^2 + g(\mathbf{x}), \tag{22}$$

where, $\mu_x \in (0, 1/L_x]$ and $\hat{\mathbf{x}}_k = \mathbf{x}_k + w \cdot (\mathbf{x}_k - \mathbf{x}_{k-1})$ with $w \in [0, 1)$. Here, we have used a momentum, also called inertial, technique to improve the convergence behavior [20]. It is straightforward to show that the solution of (22) is given by

$$\mathbf{x}_{k+1} = \mathcal{P}_{\mathcal{C}_\epsilon}\big(\hat{\mathbf{x}}_k - \mu_x \cdot \nabla_x F(\mathbf{x}_k, \mathbf{z}_{k+1})\big), \tag{23}$$

in which, $\mathcal{P}_{\mathcal{C}_\epsilon}$ denotes projection onto the set $\mathcal{C}_\epsilon$. This projection can be implemented by the algorithm proposed in [21]. The overall solver of (10) is summarized in Algorithm 1, whose convergence is guaranteed using recent results on convergence of PALM technique with inertial acceleration; see [20], [22], [23].

Algorithm 1 PALM (with inertial) for solving (10)

```
0: Inputs: $\mathbf{y}$, $\mathbf{A}$, $(\mathbf{x}_0, \mathbf{z}_0)$, $\epsilon$, $\alpha$, $w$
0: for $k = 0, 1, \cdots$ do
0:     $\mathbf{z}_{k+1} = S_{\mu_z \cdot \alpha}\big((1-\mu_z) \cdot \mathbf{z}_k + \mu_z \cdot f_\beta(\mathbf{x}_k)\big)$
0:     $\hat{\mathbf{x}}_k = \mathbf{x}_k + w \cdot (\mathbf{x}_k - \mathbf{x}_{k-1})$
0:     $\mathbf{x}_{k+1} = \mathcal{P}_{\mathcal{C}_\epsilon}\big(\hat{\mathbf{x}}_k - \mu_x \cdot \nabla_x F(\mathbf{x}_k, \mathbf{z}_{k+1})\big)$
0: end for
0: Output: $(\mathbf{x}_k, \mathbf{z}_k)$ =0
```

Now, in order to obtain an approximate solution to (7), we should solve (10) for a decreasing sequence of α, as done in standard penalty methods [18]. To this end, we consider a sequence $\{\alpha_j\}_{j \geq 1}$, where $\alpha_{j+1} = c \cdot \alpha_j$ for some $0 < c < 1$. Algorithm 2 gives a description of the overall algorithm to solve (7).

Algorithm 2 L0Soft for solving (7)

```
0: Inputs: $\mathbf{y}$, $\mathbf{A}$, $(\mathbf{x}_0, \mathbf{z}_0)$, $\epsilon$, $\alpha_1$, $w$, $c$
0: for $j = 1, 2, \cdots$ do
0:     $(\mathbf{x}_j, \mathbf{z}_j) = \text{PALM}(\mathbf{y}, \mathbf{A}, (\mathbf{x}_{j-1}, \mathbf{z}_{j-1}), \epsilon, \alpha_j, w)$
0:     $\alpha_{j+1} = c \cdot \alpha_j$
0: end for
0: Output: $\mathbf{x}_j$ =0
```

C. Implementation details

To initialize the algorithm, we set $\mathbf{x}_{-1} = \mathbf{0}$, $\mathbf{x}_0 = \mathbf{A}^\dagger \mathbf{y}$, and $\mathbf{z}_0 = f_\beta(\mathbf{x}_0)$, where $\mathbf{A}^\dagger$ denotes the Moore–Penrose pseudoinverse of $\mathbf{A}$. Moreover, we choose $\alpha_1 = 1$, which was empirically found to work well.

Additionally, for μ_z, a value close to 1, say 0.95 leads to good performance. For setting μ_x, we first prove the following lemma.

Lemma 1. *In Algorithm 1 and with the initialization of* $\mathbf{z}_0$ *as described above, we have*

$$\forall i, k: \quad |z_k^i| \leq 1, \tag{24}$$

where z_k^i *denotes the i-th entry of* $\mathbf{z}_k$*.*

Proof. First, note that $\forall x$ and $\forall \lambda, \beta \geq 0$ we have $|\mathcal{S}_\lambda(x)| \leq |x|$ and $|f_\beta(x)| \leq 1$. So, from (21) we can write

$$\forall i, k: \quad |z_{k+1}^i| \leq (1-\mu_z) \cdot |z_k^i| + \mu_z. \tag{25}$$

Then, recursive application of the above inequality yields

$$\begin{aligned} \forall i, K: \quad |z_{K+1}^i| &\leq (1-\mu_z)^K + \sum_{k=0}^{K} \mu_z \cdot (1-\mu_z)^k \\ &\leq (1-\mu_z)^K + \mu_z \frac{1-(1-\mu_z)^K}{1-(1-\mu_z)} \\ &= 1. \end{aligned} \tag{26}$$

Therefore, we can further bound L_x in (17) as $L_x \leq 5\beta^2$, and since $\mu_x \in (0, 1/L_x]$, we can deduce an admissible range for μ_x.

III. Simulation results

In this section, the performance of L0Soft is evaluated and compared with ISP-Hard [21], which is an iterative proximal-projection method for solving (1), IHT, and SL0 [17], as they all aim at solving ℓ_0 minimization, i.e., problems (1) or (2). For ISP-Hard[1] and SL0[2], we have used their available MATLAB codes. Furthermore, for IHT we have used its optimally tuned implementation[3] proposed in [24], which does not require the tuning parameter λ in (2) as an input. In addition to IHT, we have also used a more sophisticated variant of it, called two stage thresholding (TST) [24]. ISP-Hard and SL0 were run with their default parameters, which performed well. The parameters of L0Soft were chosen as $\beta = 5$, $w = 0.9$, and $c = 0.9$. Other parameters were set as suggested in Subsection II-C. Moreover, all the algorithms were run for 300 iterations, which ensured convergence. As a rough measure of computational complexity, we report average runtimes of the algorithms. Our simulations were conducted using MATLAB on a 64 bit Windows 7 operating system with 16 GB RAM and an intel core i7 CPU.

We have performed two sets of sparse recovery simulations: one on synthetic data and the other on real data. The details are given in the following subsections.

A. Synthetic data

As a common practice, we generated a sparse signal, $\mathbf{x}$, of length $n = 1000$ from a Bernoulli-Gaussian distribution. Then, we took a number of $m = 400$ random measurements by using a measurement matrix, $\mathbf{A}$, whose entries are generated according to a normal distribution. We varied the number of non-zeros entries, denoted by s, in the sparse signal. After adding zero-mean Gaussian noise with standard deviation of σ to the obtained measurement vector, we applied the competing algorithms on these noisy measurements to get an estimation of the underlying sparse signal. Each experiment

[1]https://sites.google.com/site/msaadeghii/publications
[2]http://ee.sharif.edu/~SLzero/
[3]http://sparselab.stanford.edu/OptimalTuning/code.htm

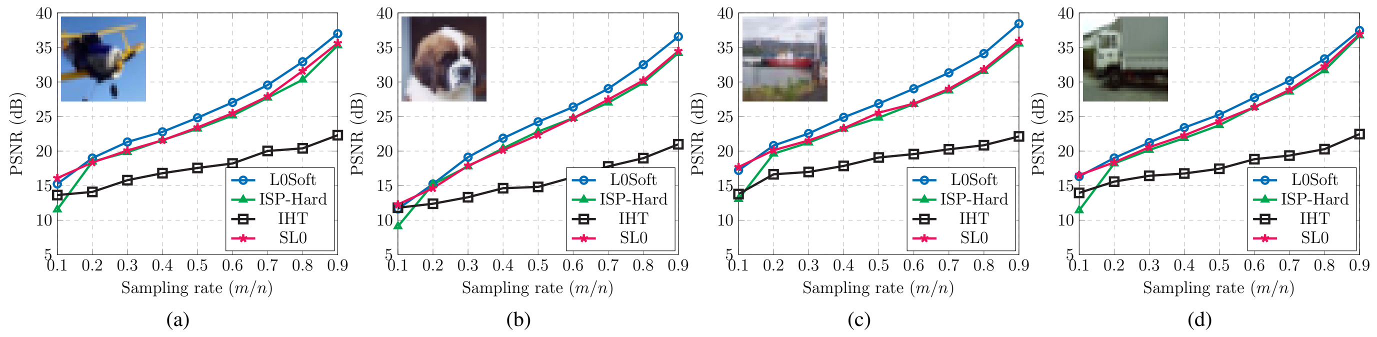

Fig. 2: Results of reconstructing some 32×32 natural images from underdetermined Gaussian measurements with different sampling ratios (m/n).

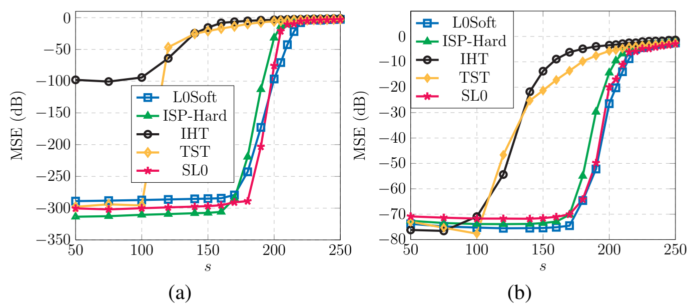

Fig. 3: Average MSEs (dB) obtained by different algorithms versus number of non-zeros (s), when recovering sparse signals of length $n = 1000$ from $m = 400$ (a) noiseless and (b) noisy ($\sigma = 0.001$) Gaussian measurements.

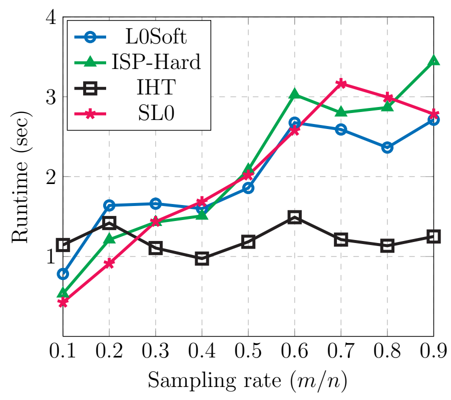

Fig. 4: Average runtimes (in second) of the algorithms.

(corresponding to a fixed s) was repeated 300 times and then the average normalized mean squared error (MSE) between the estimated signal, denoted by $\hat{\mathbf{x}}$, and the original one was computed as: $\mathrm{MSE}(\mathbf{x}, \hat{\mathbf{x}}) = 20 \log(\|\mathbf{x} - \hat{\mathbf{x}}\|_2 / \|\mathbf{x}\|_2)$.

The results, for both noiseless and noisy ($\sigma = 0.001$) recovery, and different sparsity levels are shown in Fig. 3. As can be seen, ISP-Hard, SL0, and L0Soft perform much better than IHT and TST, especially for larger values of s. Furthermore, L0Soft reaches a lower MSE than ISP-Hard and SL0 when recovering less sparse signals.

B. Real data

In the second experiment, we considered recovery of natural images from underdetermined Gaussian measurements. More precisely, let $\mathbf{X} \in \mathbb{R}^{\sqrt{n} \times \sqrt{n}}$ be a natural image, and $\mathbf{x} \in \mathbb{R}^n$ denote its vectorized version. Then, the goal of this experiment is to recover $\mathbf{x}$ from $\mathbf{y} = \mathbf{\Phi}\mathbf{x}$, where $\mathbf{\Phi} \in \mathbb{R}^{m \times n}$ with $m < n$ is a random matrix with i.i.d. entries drawn from a normal distribution. A key property of natural images utilized here is that they have sparse representation in appropriate dictionaries. That is, assuming $\mathbf{\Psi} \in \mathbb{R}^{n \times p}$ $(p > n)$ to be a well-chosen dictionary and $\mathbf{x} = \mathbf{\Psi}\mathbf{a}$, then it is expected that $\mathbf{a}$ is sparse. Therefore, the recovery problem is to estimate the sparsest solution of $\mathbf{y} = \mathbf{\Phi}\mathbf{\Psi}\mathbf{a} = \mathbf{A}\mathbf{a}$. Let the sparsest solution be $\hat{\mathbf{a}}$. Then, an estimate of the underlying image would be $\hat{\mathbf{x}} = \mathbf{\Psi}\hat{\mathbf{a}}$.

Testing images were chosen from the publicly available CIFAR-10 dataset[4], which consists of 32×32 natural images of different categories. So, here $n = 1024$. The sparsifying dictionary, i.e., $\mathbf{\Psi}$, was chosen as an overcompete DCT matrix of size 1024×4096. The competing algorithms were applied on underdetermined Gaussian measurements, $\mathbf{y} = \mathbf{\Phi}\mathbf{x}$, with different sampling ratios, defined as m/n. It should be noted that the optimally tuned iterative thresholding algorithms designed by [24] do not work here. This is because in this case the (effective) measurement matrix $\mathbf{A}$ does not have the structure assumed in [24]. So, we used IHT by manually tuning its regularization parameter, λ, for different scenarios. We computed peak signal to noise ratio (PSNR) between the original image and the estimated one. The results for some sample images are shown in Fig. 2.

Inspecting Fig. 2 reveals that ISP-Hard, SL0, and L0Soft have much better reconstruction PSNR than IHT. Furthermore, L0Soft can reconstruct images more reliably than ISP-Hard and SL0, outperforming them by $2 - 3$ dB in some cases. Figure 4 compares the runtime of the algorithms, averaged over different test images, versus sampling rate. As shown

[4]https://www.cs.toronto.edu/~kriz/cifar.html

in this figure, ISP-Hard, SL0, and L0Soft have comparable runtimes. IHT, on the other hand, has the lowest runtime, however, as observed in Fig. 2, it does not perform well in terms of PSNR.

IV. Conclusion

In this paper, we proposed a new solver for ℓ_0 minimization. The main idea is based on the fact that for a vector, the ℓ_0 norm can be computed as the ℓ_1 norm of the sign of that vector, where the sign function is applied entry-wise. Using this, the ℓ_0 minimization problem was converted to an equivalent ℓ_1 minimization problem. The new problem was then solved via proximal algorithms, leading to an iterative soft-thresholding scheme. Simulation results on recovery of synthetically generated sparse signals, as well as natural images, from underdetermined random measurements showed that the proposed algorithm performs better than some other approaches to solving ℓ_0 minimization problem.

References

[1] M. Elad, *Sparse and Redundant Representations*, Springer, 2010.

[2] J. Wright, Y. Ma, J. Mairal, G. Sapiro, T. S. Huang, and S. Yan, "Sparse representation for computer vision and pattern recognition," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 1031–1044, 2010.

[3] D. L. Donoho, "Compressed sensing," *IEEE Trans. on Information Theory*, vol. 52, no. 4, pp. 1289–1306, April 2006.

[4] E. J. Candès and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Proc. Magazine*, vol. 25, no. 2, pp. 21–30, 2008.

[5] G. Davis, S. Mallat, and M. Avellaneda, "Adaptive greedy approximations," *Constructive Approximation*, vol. 13, no. 1, pp. 57–98, 1997.

[6] J. A. Tropp and S. J. Wright, "Computational methods for sparse solution of linear inverse problems," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 948–958, 2010.

[7] S. S. Chen, D. D. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Rev.*, vol. 43, no. 1, pp. 129–159, 2001.

[8] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition," in *In Proc. Asilomar Conf. Signal Syst. Comput.*, 1993.

[9] I. Daubechies, M. Defrise, and C. De-Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Comm. Pure Appl. Math.*, vol. 57, no. 11, pp. 1413–1457, 2004.

[10] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imag. Sci.*, vol. 2, no. 1, pp. 183–202, 2009.

[11] J. M. Bioucas-Dias and M. A. T. Figueiredo, "A new twist: Two-step iterative shrinkage/thresholding algorithms for image restoration," *IEEE Transactions on Image Processing*, vol. 16, no. 12, pp. 2992–3004, 2007.

[12] D. L. Donoho, A. Maleki, and A. Montanari, "Message-passing algorithms for compressed sensing," *Proc. Nat. Acad. Sci.*, vol. 106, no. 45, pp. 18 914–18 919, 2009.

[13] M. A.T. Figueiredo, R. D. Nowak, and S. J. Wright, "Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems," *IEEE Journal of Selected Topics in Signal Processing*, vol. 1, no. 4, pp. 586–597, 2007.

[14] J. Becker, S. Bobin and E. J. Candès, "NESTA: A fast and accurate first-order method for sparse recovery," *SIAM J. Imaging Sci.*, vol. 4, no. 1, pp. 1–39, 2011.

[15] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends in Optimization*, vol. 1, no. 3, pp. 127–239, 2014.

[16] T. Blumensath and M. E. Davies, "Iterative thresholding for sparse approximations," *Journal of Fourier Analysis and Applications*, vol. 14, no. 5, pp. 629–654, 2008.

[17] H. Mohimani, M. Babaie-Zadeh, and Ch. Jutten, "A fast approach for overcomplete sparse decomposition based on smoothed ℓ^0 norm," *IEEE Trans. on Signal Processing*, vol. 57, no. 1, pp. 289–301, 2009.

[18] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer, 1999.

[19] J. Bolte, S. Sabach, and M. Teboulle, "Proximal alternating linearized minimization for nonconvex and nonsmooth problems," *Mathematical Programming*, vol. 146, no. 1, pp. 459–494, 2014.

[20] P. Ochs, Y. Chen, T. Brox, and T. Pock, "iPiano: Inertial proximal algorithm for nonconvex optimization," *SIAM J. Imag. Sci.*, vol. 7, no. 2, pp. 388–1419, 2014.

[21] M. Sadeghi and M. Babaie-Zadeh, "Iterative sparsification-projection: Fast and robust sparse signal approximation," *IEEE Trans. on Signal Proc.*, vol. 64, no. 21, pp. 5536–5548, 2016.

[22] P. Ochs, "Unifying abstract inexact convergence theorems for descent methods and block coordinate variable metric iPiano," Tech. Rep., 2016, [Available Online] arXiv:1602.07283.

[23] R. Ioan Boţ, E. R. Csetnek, and S. C. László, "An inertial forward–backward algorithm for the minimization of the sum of two nonconvex functions," *EURO Journal on Computational Optimization*, vol. 4, no. 1, pp. 3–25, 2016.

[24] A. Maleki and D. L. Donoho, "Optimally tuned iterative reconstruction algorithms for compressed sensing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 2, pp. 330–341, 2010.