

# KNOWLEDGE-AIDED NORMALIZED ITERATIVE HARD THRESHOLDING ALGORITHMS FOR SPARSE RECOVERY

Qianru Jiang<sup>\*†</sup>, Rodrigo C. de Lamare<sup>†‡</sup>, Yuriy Zakharov<sup>†</sup>, Sheng Li<sup>\*</sup> and Xiongxiang He<sup>\*</sup>

<sup>\*</sup> College of Information Engineering, Zhejiang University of Technology, Zhejiang, People's Republic of China

<sup>†</sup> Department of Electronic Engineering, University of York, York YO10 5DD, U.K

<sup>‡</sup> CETUC, PUC-Rio, Rio de Janeiro 22451-900, Brazil

jqr08141989@163.com, rcd1500@ohm.york.ac.uk, yury.zakharov@york.ac.uk, shengli@zjut.edu.cn, hxx@zjut.edu.cn

## ABSTRACT

This paper deals with the problem of sparse recovery often found in compressive sensing applications exploiting *a priori* knowledge. In particular, we present a knowledge-aided normalized iterative hard thresholding (KA-NIHT) algorithm that exploits information about the probabilities of nonzero entries. We also develop a strategy to update the probabilities using a recursive KA-NIHT (RKA-NIHT) algorithm, which results in improved recovery. Simulation results illustrate and compare the performance of the proposed and existing algorithms.

**Index Terms**— compressed sensing, iterative hard thresholding, prior information, probability estimation, sparse recovery

## 1. INTRODUCTION

Sparse signal recovery problem is one of the fundamental problems addressed in the compressed sensing (CS) [1, 2] which has been successfully utilized in many areas such as image processing [3], wireless sensor networks [4], and face recognition [5]. Most often this problem refers to recovering a sparse signal  $\mathbf{x}$  from its linear measurements:

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{e}, \quad (1)$$

where  $\|\mathbf{x}\|_0 \leq K$  (here  $\|\cdot\|_0$  counts the number of nonzero elements),  $\mathbf{A} \in \mathbb{R}^{M \times N}$  with  $M \leq N$ , and  $\mathbf{e} \in \mathbb{R}^{M \times 1}$  is the noise which is modeled by independent and identically distributed Gaussian random variables with zero mean and variance  $\sigma^2$ .

Although most of the signals are not directly sparse, they can be sparsely represented by a well-chosen dictionary  $\Psi \in \mathbb{R}^{L \times N}$  with  $L \leq N$ . A signal  $\mathbf{s}$  is said to be sparsely represented under  $\Psi$  if

$$\mathbf{s} = \Psi\mathbf{x}, \quad (2)$$

This work was partially supported by National Science Foundation of P.R. China (Grant: 61473262, and 61503339), Zhejiang National Science Foundation (Grant: LY18F010023). The work of Q. Jiang was also supported by China Scholarship Council Funding.

Typical examples of the dictionaries include the Fourier matrix for frequency-sparse signals, a multiband modulated Discrete Prolate Spheroidal Sequences (DPSSs) dictionary for sampled multiband signals [6] and one learned from the training data [7]. In CS, a sensing matrix  $\Phi \in \mathbb{R}^{M \times L}$ ,  $M \leq L$  [8, 9] is utilized to capture most of the information contained in the signal  $\mathbf{s}$  that can be sparsely represented under  $\Psi$ . Thus, for the measurement in (1) we have  $\mathbf{A} = \Phi\Psi$ . Algorithms for finding the sparse vector  $\mathbf{x}$  from the measurement  $\mathbf{y}$  include the orthogonal matching pursuit (OMP) [10], basic pursuit (BP) [11], least absolute shrinkage and selection operator (LASSO) [12], iterative hard thresholding (IHT) [13], dichotomous coordinate descent (DCD)-based algorithm [14] and  $\ell_1$ -homotopy [15].

A sparse signal can be successfully recovered when both the values of the nonzero elements of  $\mathbf{x}$  and their positions are accurately estimated. The use of *a priori* information about such values and their positions has been shown to provide improvements to the recovery task [16] - [17]. In particular, prior work [18] has exploited the probabilities of the nonzero elements to improve the recovery using the OMP algorithm. However, there has been no attempt so far to consider the improvement of other existing algorithms such as the normalized iterative hard thresholding (NIHT) [19] with the exploitation of prior knowledge. Therefore, we propose a knowledge-aided NIHT (KA-NIHT) algorithm and a recursive KA-NIHT (RKA-NIHT) algorithm, which can exploit prior information in the form of probabilities of nonzero entries. Note that the probabilities that have been used in [18] as *a priori* knowledge are fixed. However, in practice, such a precise knowledge can hardly be available. A more practical approach is to consider these probabilities just as educated guesses, and update them within the iterations using the RKA-NIHT algorithm. To this end, we propose the use of an adaptive set of probabilities and develop a recursive procedure to compute these probabilities over the iterations. Simulations show that the proposed KA-NIHT and RKA-NIHT algorithms outperform existing recovery techniques.

The organization of the remaining part of the paper is as

follows. Some preliminary work is introduced in Section 2. Section 3 describes the proposed knowledge-aided algorithms using the initial probability vector and adjustment of the probability vector through a recursive method based on the NIHT algorithm. Section 4 presents and discusses simulation results. Section 5 gives the conclusion of our work.

## 2. PRELIMINARY WORK

In compressive sensing systems, a designer often employs a recovery algorithm to obtain a sparse solution. A recovery algorithm can be formulated as an optimization problem that can be written as:

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x}: \|\mathbf{x}\|_0 \leq K} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2, \quad (3)$$

where  $\|\cdot\|_2$  denotes the 2-norm of a vector. Thus, a sparse vector  $\hat{\mathbf{x}}$  should be found so that the error  $\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2$  is minimized under the  $K$ -sparsity constraint  $\|\mathbf{x}\|_0 \leq K$ . In the following subsections, the NIHT algorithm and types of *a priori* information on the sparsity are introduced.

### 2.1. Normalized Iterative Hard Thresholding

Under the condition of the restricted isometry property [20], the sparse recovery can be obtained by using the NIHT algorithm [19]. In the NIHT algorithm, the cost function decreases in every iteration until the convergence. Let the solution vector be initialized as  $\mathbf{x}^0 = \mathbf{0}$ . The following iterations ( $l \geq 1$ ) are used in the NIHT algorithm:

$$\mathbf{x}_{l+1} = H_K(\mathbf{x}_l + \mu \mathbf{A}^T(\mathbf{y} - \mathbf{A}\mathbf{x}_l)), \quad (4)$$

where  $H_K(\cdot)$  is an operator setting all elements of a vector to zero except for the  $K$  elements with the largest magnitudes. The step size  $\mu$  is adapted to maximally minimize the error in every iteration:

$$\mu = \frac{\mathbf{g}_{\Lambda_l}^T \mathbf{g}_{\Lambda_l}}{\mathbf{g}_{\Lambda_l}^T \mathbf{A}_{\Lambda_l}^T \mathbf{A}_{\Lambda_l} \mathbf{g}_{\Lambda_l}}, \quad (5)$$

where  $\Lambda_l$  is the support of  $\mathbf{x}_l$  and  $\mathbf{g} = \mathbf{A}^T(\mathbf{y} - \mathbf{A}\mathbf{x}_l)$  is the negative gradient of  $\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2$ .

### 2.2. *A priori* information on sparsity

The NIHT algorithm exploits *a priori* information on the sparsity of the solution in the form of the number of non-zero elements  $K$ . The performance of a sparse recovery algorithm can be improved if extra *a priori* information on the sparsity of the solution is available. Let a sparse signal  $\mathbf{x} = [x_1, \dots, x_N]^T$  be generated using the following rule. The  $i$ -th entry of  $\mathbf{x}$  is given by

$$x_i = w_i \vartheta_i, \quad (6)$$

where  $w_i$  is a deterministic non-zero value, and  $\vartheta_i \in \{0, 1\}$  is a random binary variable used to decide whether the entry is zero or non-zero according to a probability distribution. The probability for  $\vartheta_i = 1$  is denoted as  $p_i$ , and the random variable  $\{\vartheta_i\}_{i=1}^N$  are assumed independent. The support of  $\mathbf{x}$  is given by  $\Lambda = \{i | \vartheta_i = 1\}$ . In [18], OMP [10], BP [11] and LASSO [12] algorithms are extended to exploit *a priori* information in the form of the probability distribution  $\mathbf{p} = [p_1, \dots, p_N]^T$  which are named as Log-Weighted OMP (LW-OMP), Log-Weighted BP (LW-BP), Log-Weighted LASSO (LW-LASSO), respectively. The results show that excellent sparse recovery can be achieved if the probability distribution is sufficiently non-uniform, especially using LW-OMP algorithm. Therefore below we will use it as a benchmark for comparison with our algorithm.

The goal of this paper is to propose an efficient algorithm to estimate  $\mathbf{x}$  from the observation  $\mathbf{y}$  using  $\mathbf{p} = [p_1, \dots, p_N]^T$  as prior probabilities based on the NIHT algorithm.

## 3. PROPOSED RECONSTRUCTION ALGORITHMS

The NIHT algorithm according to (4) selects the support elements with the highest magnitudes. With such strategy, it is possible that non-zero elements with low magnitudes are missed. By exploiting the prior probabilities  $\mathbf{p} = [p_1, \dots, p_N]^T$  the support estimate can be made more accurate and consequently the performance of sparse recovery improved.

We propose to recover the support in every iteration of the NIHT algorithm using the following operation:

$$\Lambda = \text{sup}[S_K(|\mathbf{x} + \mu \mathbf{g}| + \alpha \log(\mathbf{p}))], \quad (7)$$

where  $S_K(\cdot)$  is an operator to sort elements of a vector in descending order, sets last  $(N - K)$  elements to zero, and return the elements back to the original order. The operator  $\text{sup}[\cdot]$  extracts positions of non-zero elements. Note that the notation  $|\mathbf{x} + \mu \mathbf{g}|$  means element-wise magnitudes of the vector elements. Similarly,  $\log(\mathbf{p})$  means log of each element of vector  $\mathbf{p}$ . The term  $\log(\mathbf{p})$  introduces a penalty in the iterations. If  $\alpha = 0$ , then we arrive at the NIHT algorithm. If  $\alpha > 0$ , then for a high  $p_i$  (close to one) the penalty is small, while for a small  $p_i$  the penalty is high. As a result, if it is *a priori* known that the probability of appearance of the  $i$ -th element  $p_i$  is low, it is unlikely that it would be selected into the support  $\Lambda$ . The parameter  $\alpha$  controls the tradeoff between the importance of the current magnitude  $|x_i + \mu g_i|$  of an element and its *a priori* probability  $p_i$ . The modification of the NIHT algorithm that exploits the support estimate according to (7) is named KA-NIHT.

Note that in the KA-NIHT algorithm, the *a priori* probabilities  $\mathbf{p}$  are fixed. However, in practice, such a precise knowledge can hardly be available. A more practical approach is to consider these probabilities just as educated

guesses, and update them within the iterations using the RKA-NIHT algorithm as described below.

In the RKA-NIHT algorithm, the support estimate (7) is replaced with the estimate:

$$\Lambda = \sup[S_K(|\mathbf{x} + \mu\mathbf{g}| + \alpha \log(\mathbf{q}))], \quad (8)$$

when the vector  $\mathbf{q} = \mathbf{p}$  at the first iteration and it is updated recursively at the following iterations as:

$$\mathbf{q}_\Lambda \leftarrow \mathbf{q}_\Lambda + \beta \mathbf{p}_\Lambda. \quad (9)$$

Here we denote  $\mathbf{q}_\Lambda$  a vector which is obtained from the vector  $\mathbf{q}$  by only keeping elements on the support  $\Lambda$ . The parameter  $\beta$  is a tuning parameter which is selected according to the experiments.

The steps of the RKA-NIHT algorithm are summarized as follows.

---

#### Algorithm: RKA-NIHT

---

**Input:** Data vector  $\mathbf{y} \in \mathfrak{R}^{M \times 1}$ , compressed matrix  $\mathbf{A} \in \mathfrak{R}^{M \times N}$ , the sparsity  $K$  is known. The probability vector is  $\mathbf{p}$ . The iteration number is  $I$ .  $c, \kappa, \beta$  are constants.

**Initialization:** the solution vector  $\mathbf{x}_0 = \mathbf{0}$ , and

$$\mathbf{x}_1 = H_K(\mathbf{A}^T \mathbf{y}),$$

$\Lambda_1 = \sup(\mathbf{x}_1), l = 1; \mathbf{q} \leftarrow \mathbf{p}$ .

**Repeat** until  $l > I$ :

Step 1:  $\mathbf{g}_l = \mathbf{A}^T(\mathbf{y} - \mathbf{A}\mathbf{x}_l)$

Step 2:  $\mu_l = \frac{\mathbf{g}_{\Lambda_l}^T \mathbf{g}_{\Lambda_l}}{\mathbf{g}_{\Lambda_l}^T \mathbf{A}_{\Lambda_l}^T \mathbf{A}_{\Lambda_l} \mathbf{g}_{\Lambda_l}}$

Step 3:  $\Lambda_{l+1} = \sup[S_K(|\mathbf{x}_l + \mu_l \mathbf{g}_l| + \alpha \log(\mathbf{q}))]$

Step 4:  $\hat{\mathbf{x}}_{l+1} = (\mathbf{x}_l + \mu_l \mathbf{g}_l)_{\Lambda_{l+1}}$

Step 5: **If**  $\Lambda_{l+1} = \Lambda_l$  **then**  $\mathbf{x}_{l+1} = \hat{\mathbf{x}}_{l+1}$   
**else**  $\varrho = (1-c)(\|\hat{\mathbf{x}}_{l+1} - \mathbf{x}_l\|_2^2) / (\|\mathbf{A}(\hat{\mathbf{x}}_{l+1} - \mathbf{x}_l)\|_2^2)$   
**if**  $\mu_l \geq \varrho$  **repeat** until  $\mu_l \leq \varrho$   
 $\mu_l \leftarrow \mu_l / (\kappa(1-c))$   
 $\Lambda_{l+1} = \sup[S_K(|\mathbf{x}_l + \mu_l \mathbf{g}_l| + \alpha \log(\mathbf{q}))]$   
 $\hat{\mathbf{x}}_{l+1} = (\mathbf{x}_l + \mu_l \mathbf{g}_l)_{\Lambda_{l+1}}$   
 $\mathbf{x}_{l+1} = \hat{\mathbf{x}}_{l+1}$

Step 6: Update  $\mathbf{q}$ :

$$\mathbf{q}_{\Lambda_{l+1}} \leftarrow \mathbf{q}_{\Lambda_{l+1}} + \beta \mathbf{p}_{\Lambda_{l+1}}$$

Step 7:  $l = l + 1$

**Output:**  $\hat{\mathbf{x}} = \mathbf{x}_l$

---

The KA-NIHT algorithm is the particular case of the algorithm RKA-NIHT when step 6 is removed.

The computational complexity of NIHT, KA-NIHT and RKA-NIHT per iteration is calculated and shown in Table 1.

#### 4. SIMULATION RESULTS

In this section, we investigate the performance of OMP [10], LW-OMP [18], NIHT [19], KA-NIHT, RKA-NIHT and Oracle NIHT algorithms for various  $M$  and noise variance  $\sigma^2$  with  $N = 240$ , and  $S = 1000$  simulation trials.

The values  $\{w_i\}_{i=1}^N$  in (6) are drawn from a Gaussian distribution  $N(0, 1)$ . The support probabilities are modeled as follows.

Elements of  $\mathbf{x}$  are divided into  $G$  groups. The proportions  $h_n$  of elements in the groups are different, but  $\sum_{n=1}^G h_n = 1$ . Every group contains  $N_n = h_n N$  elements whose support probabilities are equal to  $\bar{p}_n$ . For a sparse signal with support  $\Lambda$ , the sparsity  $K$  is the number of non-zero elements  $K = |\Lambda|$ . The average of  $|\Lambda|$  is denoted as  $\bar{K} = \sum_{i=1}^N p_i = \sum_{n=1}^G N_n \bar{p}_n$ . In our simulation, the sparse vector is divided into  $G = 4$  groups with  $N_1 = 210, N_2 = 20, N_3 = 5, N_4 = 5, N_n \bar{p}_n = 4$ , and  $\bar{K} = 16$ . That is, each group has on average four non-zero coefficients.

We compare the true support  $\Lambda_t$  with its estimate  $\hat{\Lambda}_t$  in the  $t$ -th simulation trail using

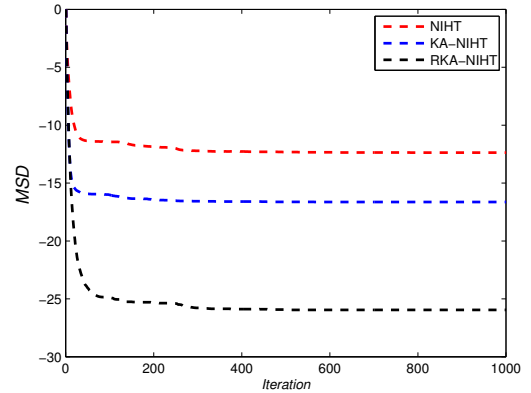
$$P_{\text{recovered}} = \frac{1}{S} \sum_{t=1}^S \frac{|\Lambda_t \cap \hat{\Lambda}_t|}{|\Lambda_t|}, \quad (10)$$

Another performance measure adopted in this paper is the Mean Square Deviation (MSD) defined as:

$$MSD = \frac{1}{S} \sum_{t=1}^S \frac{\|\mathbf{x}^{(t)} - \hat{\mathbf{x}}^{(t)}\|_2^2}{\|\mathbf{x}^{(t)}\|_2^2}, \quad (11)$$

where  $\mathbf{x}^{(t)}$  is the true vector and  $\hat{\mathbf{x}}^{(t)}$  is the recovered vector in the  $t$ -th simulation trial.

Fig. 1 draws the MSD evolution with iterations in the three NIHT based algorithms. The variance of noise is  $\sigma^2 = 10^{-3}$ . The dimension of  $\mathbf{y}$  is  $M = 70$ . The other parameters are set to:  $c = 0.01, \kappa(1-c) = 2, \beta = 0.6, \alpha = \frac{1.5}{\sum_{i=1}^N q_i}$ . It can be seen that the proposed algorithms outperform the original NIHT algorithm, with the RKA-NIHT algorithm providing the best performance.



**Fig. 1.** The MSD versus iteration number, with  $M = 70$ ;

Fig. 2 (a) demonstrates the proportion of coefficients recovered in six methods with the dimension of the vector  $\mathbf{y}$  varying from  $M = 40$  to  $M = 80$ . Fig. 2 (b) shows the MSD performance.

We can conclude the following:

The proportion of correct recovery increases with the larger dimension of  $\mathbf{y}$ .

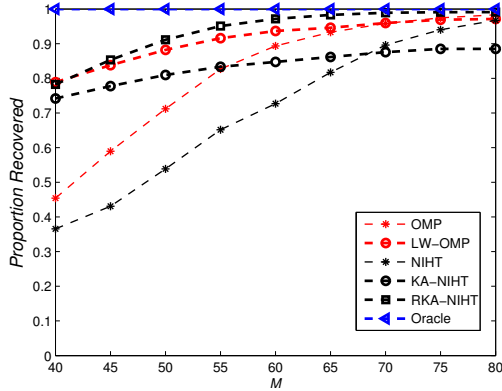
The recovery algorithms with prior information perform better than the original recovery algorithms such as the LW-OMP against OMP and KA-NIHT against NIHT.

As for the KA-NIHT and RKA-NIHT algorithms, the performance of the scheme with adjustable probabilities is better than that with fixed probabilities.

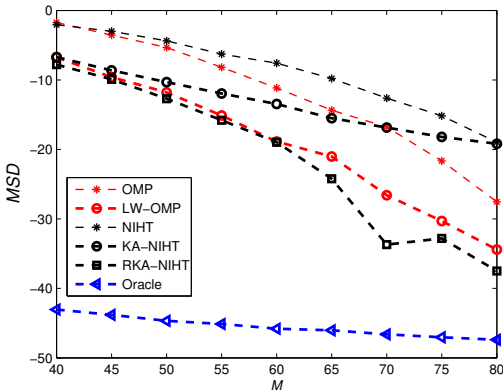
**Table 1.** Computational complexity of three algorithms per iteration.

	Additions	Multiplications	Comparison
NIHT	$KM + K + 2N + 3MN + M - 5$	$KM + K + 2N + 3MN + 2M + 3$	$2N \log N + 1$
KA-NIHT	$KM + K + 5N + 2.75MN - 2.5$	$KM + K + 2.5N + 2.75MN + 1.75M + 0.75$	$2.75N \log N + 1.75K + 0.75$
RKA-NIHT	$KM + 2K + 2.6N + 2.4MN - 1.8$	$KM + 2K + 1.8N + 2.4MN + 1.4M + 0.4$	$2.75N \log N + 1.75K + 0.75$

The RKA-NIHT algorithm shows a better performance than the LW-OMP algorithm, also exploiting the *a priori* probabilities.



(a)



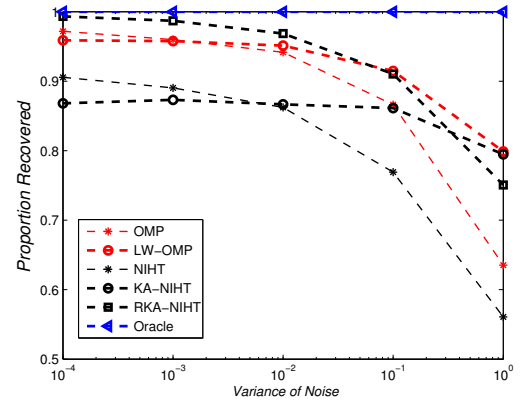
(b)

**Fig. 2.** The proportion of coefficients recovered and MSD versus sensing matrix dimension  $M$ .

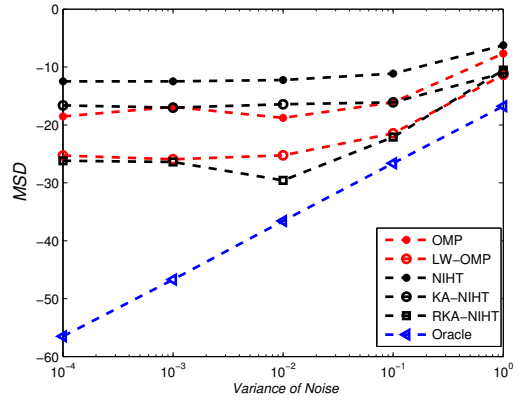
Fig. 3 shows the algorithm performance for varying levels of noise. The length of  $y$  is  $M = 70$ . Other parameters are the same as for Fig. 2. Results in Fig. 3 demonstrate that the proposed RKA-NIHT algorithm outperforms the other algorithms in the large range of the noise level.

## 5. CONCLUSION

In this paper, we have developed knowledge-aided normalized iterative hard thresholding algorithms for sparse recovery problems. The proposed algorithms, named KA-NIHT and RKA-NIHT algorithms, have been designed considering the initial probabilities which are



(a)



(b)

**Fig. 3.** The proportion of coefficients recovered and MSD versus variance of noise  $\sigma^2$ .

given at the start as prior knowledge and a recursive procedure to update the probabilities of the non-zero coefficients, respectively. The use of prior probabilities can improve the accuracy of finding the positions for non-zero elements. The simulations have shown that the proposed KA-NIHT and RKA-NIHT algorithms perform very well as compared to existing algorithms.

## 6. REFERENCES

- [1] D. L. Donoho, "Compressed sensing," *IEEE Transactions on information theory*, vol. 52, no. 4, pp. 1289–1306, 2006.
- [2] E. J. Candès and M. B. Wakin, "An introduction to compressive sampling," *IEEE signal processing magazine*, vol. 25, no. 2, pp. 21–30, 2008.
- [3] G. Li, X. Li, S. Li, H. Bai, Q. Jiang, and X. He, "Designing robust sensing matrix for image compression," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5389–5400, 2015.
- [4] S. Xu, R. C. de Lamare, and H. V. Poor, "Distributed compressed estimation based on compressive sensing," *IEEE Signal Processing Letters*, vol. 22, no. 9, pp. 1311–1315, 2015.
- [5] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 2, pp. 210–227, 2009.
- [6] Z. Zhu and M. B. Wakin, "Approximating sampled sinusoids and multiband signals using multiband modulated dpss dictionaries," *Journal of Fourier Analysis and Applications*, pp. 1–48, 2016.
- [7] M. Aharon, M. Elad, and A. Bruckstein, "rmk-svd: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on signal processing*, vol. 54, no. 11, pp. 4311–4322, 2006.
- [8] G. Li, Z. Zhu, D. Yang, L. Chang, and H. Bai, "On projection matrix optimization for compressive sensing systems," *IEEE Transactions on Signal Processing*, vol. 61, no. 11, pp. 2887–2898, 2013.
- [9] Q. Jiang, S. Li, H. Bai, R. C. de Lamare, and X. He, "Gradient-based algorithm for designing sensing matrix considering real mutual coherence for compressed sensing systems," *IET Signal Processing*, vol. 11, no. 4, pp. 356–363, 2017.
- [10] Y. C. Pati, R. Rezaifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *Proc. 27th Asilomar Conf. Signals Syst. Comput.*, 1993, pp. 40–44.
- [11] E. J. Candès and T. Tao, "Decoding by linear programming," *IEEE transactions on information theory*, vol. 51, no. 12, pp. 4203–4215, 2005.
- [12] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.
- [13] T. Blumensath and M. E. Davies, "Iterative hard thresholding for compressed sensing," *Applied and computational harmonic analysis*, vol. 27, no. 3, pp. 265–274, 2009.
- [14] Y. V. Zakharov, V. H. Nascimento, R. C. De Lamare, and F. G. D. A. Neto, "Low-complexity dcd-based sparse recovery algorithms," *IEEE Access*, vol. 5, pp. 12737–12750, 2017.
- [15] D. L. Donoho and Y. Tsaig, "Fast solution of  $\ell_1$ -norm minimization problems when the solution may be sparse," *IEEE Transactions on Information Theory*, vol. 54, no. 11, pp. 4789–4812, 2008.
- [16] C. J. Miosso, R. Borries, and J. H. Pierluissi, "Compressive sensing with prior information: Requirements and probabilities of reconstruction in  $\ell_1$ -minimization," *IEEE Transactions on Signal Processing*, vol. 61, no. 9, pp. 2150–2164, 2013.
- [17] J. F. C. Mota, N. Deligiannis, and M. R. D. Rodrigues, "Compressed sensing with prior information: Optimal strategies, geometry, and bounds," *arXiv preprint arXiv:1408.5250*, 2014.
- [18] J. Scarlett, J. S. Evans, and S. Dey, "Compressed sensing with prior information: Information-theoretic limits and practical decoders," *IEEE Transactions on Signal Processing*, vol. 61, no. 2, pp. 427–439, 2013.
- [19] T. Blumensath and M. E. Davies, "Normalized iterative hard thresholding: Guaranteed stability and performance," *IEEE Journal of selected topics in signal processing*, vol. 4, no. 2, pp. 298–309, 2010.
- [20] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on information theory*, vol. 52, no. 2, pp. 489–509, 2006.