2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA) | 978-1-6654-4337-1/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICMLA52953.2021.00262

Data-driven Support Recovery for Sparse Signals with Non-stationary Modulation

Youye Xie, Michael B. Wakin, and Gongguo Tang
Department of Electrical Engineering, Colorado School of Mines, Golden, CO, USA
{youyexie, mwakin, gtang}@mines.edu

***Abstract*—Estimating a sparse signal from its low-dimensional observations arises in many applications including signal demixing and compression. If each dictionary atom undergoes an unknown modulation process, this problem becomes a sparse recovery and blind demodulation problem. In this paper, we further allow the modulation process to be different for different dictionary atoms, which is known as non-stationary modulation. In the presence of noise, the sparse signal and modulation parameters cannot be recovered exactly. We propose to solve the support recovery problem with non-stationary modulation via an optimization-inspired data-driven method. Specifically, by assuming the modulating signals live in a known common subspace and applying the lifting technique, we formulate the support recovery problem as recovering a column-wise sparse matrix from linear observations, which could then be solved via a block ℓ_1 norm regularized quadratic minimization. By unfolding the proximal gradient descent algorithm for that regularized quadratic minimization and replacing the proximal operator with a proximal network, we construct a novel recurrent neural network (RNN) to efficiently solve the support recovery problem. Experiments indicate that the proposed network is very efficient in solving the support recovery problem, can be adaptive to different sensing processes without retraining the network, and is applicable when the matrix of interest is not strictly column-wise sparse and when we only know an approximation of the sensing process.**

***Index Terms*—Machine learning, support recovery, blind demodulation, proximal gradient descent, compressed sensing**

I. Introduction

Optimization methods [1]–[3] have achieved great success in signal recovery by exploiting the signal intrinsic properties like sparsity [4], [5] and low-rankness [6], [7]. For many optimization methods, we are able to study their statistical performance [8], [9], e.g., by characterizing a sufficient number of observations and conditions on the regularization parameters for exact recovery. However, advances in data acquisition lead to high-dimensional problems [10], [11] that cannot be solved efficiently by optimization methods, and in many applications, the complicated sensing process cannot be properly modeled by the optimization program [12]. Data-driven methods [13], [14], especially deep learning methods [10], [15], [16], have emerged as a promising alternative. However, the black box nature of deep learning methods is a barrier to designing more effective network structures and limits the adoption of deep learning methods in applications where interpretability is important. Thus, the synergistic integration [12], [13], [17], [18] of traditional optimization methods and deep learning techniques has attracted great interest and holds the potential to significantly accelerate and improve recovery from complicated sensing processes.

In this paper, we apply an optimization-inspired and physics-informed data-driven method to the support recovery problem for a sparse signal that has undergone non-stationary modulation. In the following sections, we first describe the physics of the non-stationary sensing process mathematically and provide a case study on applying the physical signal model to the frequency estimation problem for damped sinusoids. Then, inspired by a traditional optimization algorithm, proximal gradient descent [19], and leveraging the knowledge of the physics of the sensing process, we propose a novel recurrent neural network (RNN) to address the support recovery problem. In experiments, we examine its efficiency and robustness under different conditions.

A. Support Recovery for Sparse Signals with Non-stationary Modulation

Estimating a sparse signal from low-dimensional observations arises in many applications such as super-resolution [20] and image compression [1]. Mathematically, after modulation, the system observes $\boldsymbol{y} = \mathbf{D}\mathbf{A}\boldsymbol{c} \in \mathbf{R}^N$, where $\boldsymbol{y} \in \mathbf{R}^N$ is the observed signal vector, $\mathbf{D} \in \mathbf{R}^{N\times N}$ is a diagonal modulation matrix, $\mathbf{A} \in \mathbf{R}^{N\times M}$ $(N < M)$ is a dictionary matrix, and $\boldsymbol{c} \in \mathbf{R}^M$ is the sparse signal vector of interest. Since $\mathbf{D}$ performs element-wise multiplication, known as modulation in signal processing, when $\mathbf{A}$ is known, recovering $\boldsymbol{c}$ and $\mathbf{D}$ from the observed $\boldsymbol{y}$ is referred to as sparse recovery and blind demodulation [1], [9], [21].

In this paper, we further generalize the model by allowing each dictionary atom to undergo a distinct modulation process; such non-stationary modulation is studied in [21], [22]. Namely, the observation vector has the form

$$\boldsymbol{y} = \sum_{j=1}^{M} c_j \mathbf{D}_j \boldsymbol{a}_j + \boldsymbol{n} \in \mathbf{R}^N, \tag{1}$$

where c_j is the j-th entry of $\boldsymbol{c}$, $\boldsymbol{a}_j$ is the j-th column of the dictionary $\mathbf{A}$, and $\boldsymbol{n}$ denotes the noise. Moreover, we assume that only J $(J < M)$ of the coefficients c_j are non-zero and the modulating signals live in a known and common subspace:

$$\mathbf{D}_j = \mathrm{diag}(\mathbf{B}\boldsymbol{h}_j),$$

This work was supported by NSF grant CCF–1704204.

978-1-6654-4337-1/21/$31.00 ©2021 IEEE
DOI 10.1109/ICMLA52953.2021.00262

where $\mathbf{B} \in \mathbf{R}^{N\times K}$ $(N > K)$ is a known, orthonormal subspace matrix and $\boldsymbol{h}_j \in \mathbf{R}^K$ is the unknown coefficient vector. In this case, recovering $\boldsymbol{h}_j$ is equivalent to recovering $\mathbf{D}_j$. A similar subspace assumption for the modulating signal can be found in the deconvolution and demixing literature [23], [24].

B. The $\ell_{2,1}$ norm Regularized Quadratic Minimization

In order to recover $\boldsymbol{c}$ and $\boldsymbol{h}$ by using the lifting technique based on Proposition 1 in [25], we can construct a column-wise sparse matrix $\mathbf{X} = [c_1\boldsymbol{h}_1 \quad c_2\boldsymbol{h}_2 \quad \cdots \quad c_M\boldsymbol{h}_M] \in \mathbf{R}^{K\times M}$ containing all the unknown parameters. In terms of the constructed unknown matrix $\mathbf{X}$, the observation (1) can be represented as $\boldsymbol{y} = \boldsymbol{\Phi}\cdot \text{vec}(\mathbf{X})$ where

$$\boldsymbol{\Phi} = [\phi_{1,1}, \cdots, \phi_{K,1}, \cdots, \phi_{1,M}, \cdots, \phi_{K,M}] \in \mathbf{R}^{N\times KM},$$

$\phi_{i,j} = \text{diag}(\boldsymbol{b}_i)\boldsymbol{a}_j \in \mathbf{R}^{N\times 1}$, and $\boldsymbol{b}_i$ is the i-th column of $\mathbf{B}$.

In presence of noise, the observations become $\boldsymbol{y} = \boldsymbol{\Phi}\cdot \text{vec}(\mathbf{X}) + \boldsymbol{n}$. In this paper, we assume each entry of $\boldsymbol{n}$ is drawn from the Gaussian distribution and denote the ground-truth matrix as $\mathbf{X}_0$. Due to the additive noise, we cannot recover $\mathbf{X}_0$ exactly. Thus we aim to recover the indices of the non-zero columns (support) in $\mathbf{X}_0$ which is equivalent to the support of the sparse signal $\boldsymbol{c}$ if we assume there is no null modulation, $\mathbf{D}_j = 0$. In order to recover the support of $\mathbf{X}_0$ from $\boldsymbol{y}$, [22], [26] propose to apply the block ℓ_1 ($\ell_{2,1}$) norm regularized quadratic minimization

$$\underset{\mathbf{X}\in\mathbf{R}^{K\times M}}{\text{minimize}} \frac{1}{2}||\boldsymbol{y} - \boldsymbol{\Phi}\cdot\text{vec}(\mathbf{X})||_2^2 + \lambda||\mathbf{X}||_{2,1}, \tag{2}$$

where the $\ell_{2,1}$ norm is defined as $||\mathbf{A}||_{2,1} = \sum_{j=1}^{M}||\boldsymbol{a}_j||_2$ and the value of λ for exact support recovery is derived in [22]. Equivalently, (2) can be written as

$$\underset{\mathbf{X}\in\mathbf{R}^{K\times M}}{\text{minimize}} \frac{1}{2}||\boldsymbol{y} - \boldsymbol{\Phi}\cdot\text{vec}(\mathbf{X})||_2^2 + \lambda\sum_{j=1}^{M}||\boldsymbol{x}_j||_2, \tag{3}$$

where $\boldsymbol{x}_j$ is the j-th column of $\mathbf{X}$.

Proximal gradient descent [19] can be applied to solve (3), using gradient descent and proximal operator steps that run sequentially and iteratively. By unfolding proximal gradient descent and applying a proximal network with the skip connection, we propose a novel RNN in Section II to efficiently solve the support recovery problem for a sparse signal with non-stationary modulation.

C. Case Study for Damped Frequency Estimation

In this section, we describe how our signal model introduced in (1) can be applied to the frequency estimation problem for damped sinusoids which naturally appears in structural health monitoring [27] and fault detection [7]. In fault detection for induction machines [28], when a fault occurs, a damped frequency component is generated and locating this frequency is essential for machine monitoring. Estimating the frequency is made more complicated due to the damping [17]. Mathematically, after discretizing the normalized frequency, θ, onto M uniformly separated grid points in $[0, 1]$, the observed signal has the form

$$\boldsymbol{y} = \sum_{j=1}^{M} c_j\mathbf{D}_j\boldsymbol{a}(\theta_j) + \tilde{\boldsymbol{n}} \in \mathbf{R}^{N\times 1}, \tag{4}$$

where c_j denotes the signal magnitude, $\boldsymbol{a}(\theta_j)$ is a sinusoidal atom with normalized frequency θ_j, $\mathbf{D}_j$ contains the damping signal $e^{-\alpha_j\cdot n}$ for $n = \{0, 1, \cdots, N-1\}$ in its diagonal entries, and $\tilde{\boldsymbol{n}}$ denotes additive noise.

By stacking damped (real) exponential samples with the damping parameter on a fine grid into a matrix and applying the singular value decomposition (SVD), we can construct a subspace matrix $\mathbf{B}$ using the left singular vectors to approximate the damping signal. Namely, we have $\mathbf{D}_j = \text{diag}(\mathbf{B}\boldsymbol{h}_j)$ and

$$\boldsymbol{y} = \sum_{j=1}^{M} c_j\,\text{diag}(\mathbf{B}\boldsymbol{h}_j)\boldsymbol{a}(\theta_j) + \boldsymbol{n} \in \mathbf{R}^{N\times 1},$$

where $\boldsymbol{n}$ consists of both the additive noise, $\tilde{\boldsymbol{n}}$, and the subspace approximation error. In this case, the support of the lifted unknown matrix $\mathbf{X} = [c_1\boldsymbol{h}_1 \quad c_2\boldsymbol{h}_2 \quad \cdots \quad c_M\boldsymbol{h}_M] \in \mathbf{R}^{K\times M}$ indicates the active frequency components. The signal model studied in this paper can also be applied to other applications such as directional of arrival estimation [21], CDMA systems [9], and nuclear magnetic resonance spectroscopy [20], [29].

D. Related Work

Most sparse recovery and blind demodulation literature [9], [24], [30]–[32] assumes a common modulation matrix $\mathbf{D}_j = \mathbf{D}$ for each dictionary atom. Specifically, [24] assumes a common modulation matrix and a dictionary consisting of complex sinusoids over a continuous frequency range. [20] generalizes the problem in [24] to accommodate non-stationary modulation. However, that paper makes a random 'sign' assumption on $\boldsymbol{h}_j$ which makes it difficult to consider the noise. In addition, [9] assumes a common modulation matrix but considers random Gaussian and Fourier dictionaries. [21], [25] extend the work of [9] by introducing non-stationary modulation with bounded noise. The support recovery problem with non-stationary modulation and unbounded Gaussian noise is studied in [22], [26] and is also the problem we study in this paper. While [22], [26] analyze and study the support recovery problem from the optimization perspective, however, this paper studies this problem from an optimization-inspired data-driven perspective.

Data-driven deep learning has achieved competitive performance in signal processing [12], [33], [34], and in particular, there are many deep networks proposed for the sparse recovery problem [12]–[14], [35], [36]. Compared to them, in this paper, we take non-stationary modulation into account. By unfolding the proximal gradient descent algorithm, we propose a novel RNN to solve the support recovery problem for a sparse signal with non-stationary modulation. The unfolding data-driven approach for signal processing was pioneered in [13]

and has been applied to many signal processing problems including matrix factorization [37], [38] and non-negative sparse recovery [12]. The unfolded version of proximal gradient descent has been mainly investigated in imaging inverse problems [39]–[41].

The rest of the paper is organized as follows. In Section II, we present our proposed RNN for support recovery of a sparse signal with non-stationary modulation. Numerical simulations are conducted in Section III to analyze the performance of the proposed approach and compare to the optimization method. Finally, we conclude this paper in Section IV.

II. Proposed Recurrent Neural Network

As introduced in Section I-B, $\ell_{2,1}$ norm regularized quadratic minimization can be applied to recover the support of the signal of interest $\mathbf{X}$. Proximal gradient descent [19] for solving (2) can be viewed as a two step iterative algorithm. The first step implements gradient descent with respect to the data fidelity, $\frac{1}{2}||\boldsymbol{y} - \boldsymbol{\Phi} \cdot \text{vec}(\mathbf{X})||_2^2$, and the second step runs a proximal operator to impose the regularization, $\lambda||\mathbf{X}||_{2,1}$. Mathematically, the proximal gradient descent iteration has the form

$$\begin{aligned}\text{vec}(\mathbf{X}^{k+1}) &= P\left[\text{vec}(\mathbf{X}^k) - \eta\boldsymbol{\Phi}^T(\boldsymbol{\Phi} \cdot \text{vec}(\mathbf{X}^k) - \boldsymbol{y})\right] \\ &= P\left[(\mathbf{I} - \eta\boldsymbol{\Phi}^T\boldsymbol{\Phi}) \cdot \text{vec}(\mathbf{X}^k) + \eta\boldsymbol{\Phi}^T\boldsymbol{y}\right],\end{aligned}$$

where k is the iteration number, η is the gradient descent step size, and

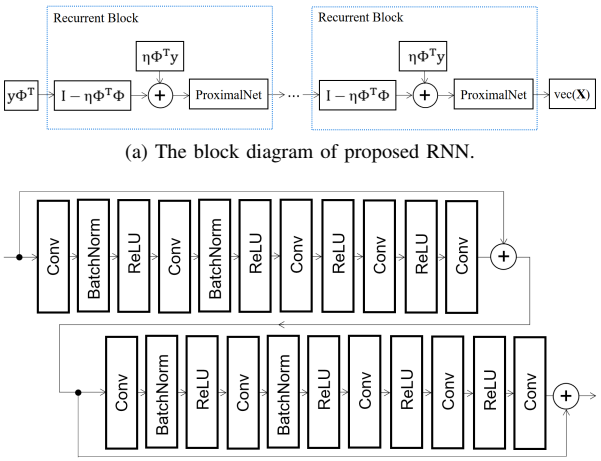
$$P[\text{vec}(\mathbf{Z})] = \arg\min_{\text{vec}(\mathbf{X})} \frac{1}{2}||\text{vec}(\mathbf{X}) - \text{vec}(\mathbf{Z})||_2^2 + \lambda||\mathbf{X}||_{2,1} \tag{5}$$

is the proximal operator.

By unfolding the proximal gradient descent and replacing the proximal operator with a proximal network with the skip connection, we construct a recurrent neural network (RNN) for support recovery shown in Fig. 1. We use $\boldsymbol{\Phi}^T\boldsymbol{y}$ as the network initial input. In the data fidelity gradient descent step, the sensing matrix $\boldsymbol{\Phi}$ and observation $\boldsymbol{y}$ are associated with a learnable step size η. The proximal operator is replaced by a proximal network consisting of convolutional layers, batch normalization layers, and ReLU layers with skip connections. All recurrent blocks share the same weights. Intuitively, the convolutional layer is responsible for analyzing the signal and combining with the ReLU layer to impose the column-wise sparsity prior. The batch normalization layer aims to improve the stability and convergence rate of the network for training. Those network layers are also found useful in deep architectures for other signal inverse problems [14], [36].

III. Numerical Simulations

We conduct several numerical simulations to evaluate the performance of our proposed RNN for the support recovery problem. We set the system parameters based on [22], such that $\ell_{2,1}$ regularized quadratic minimization could recover the support exactly with an overwhelming probability. Specifically, we set $J = K = 2$, $M = 120$, $N = 100$, and the mean and standard deviation of the Gaussian noise to



(a) The block diagram of proposed RNN.

(b) The proximal network (ProximalNet).

Fig. 1: The proposed recurrent neural network (RNN) for support recovery. In the proposed RNN, all recurrent blocks share the same weights. In the proximal network, 'Conv', 'BatchNorm', and 'ReLU' denote the convolutional layer, batch normalization layer, and the Rectified Linear Unit (ReLU) layer respectively.

0 and $\sigma = 0.1$ respectively. The entries of $\mathbf{A}$ are drawn from the standard normal distribution, and we generate a second random matrix using the standard normal distribution and orthogonalize its columns to construct $\mathbf{B}$. The J indices of non-zero columns (support) in $\mathbf{X}_0$ are uniformly selected from $\{1, 2, \cdots, 120\}$. The non-zero entries of $\mathbf{X}_0$ have the form $\text{sign}(x) + x$ where x follows the standard normal distribution and $\text{sign}(x) = -1$ when $x < 0$ and $\text{sign}(x) = 1$ for $x > 0$. Moreover, we scale $\mathbf{X}_0$ such that $\gamma = \frac{\gamma_0}{\min_{j\in T}||\boldsymbol{x}_{0,j}||_2} = 0.1$ to ensure the support recovery problem is theoretically solvable with an overwhelming probability [22], where $\gamma_0 = \sqrt{\sigma^2\mu_{max}^2 K\left[\log(M-J) + \log(N)\right]}$ and $\mu_{max} = \max_{i,j}\sqrt{N}|\mathbf{B}_{ij}|$.

Following the process above, we obtain $\boldsymbol{y}$ following (1) and generate 16000 $(\boldsymbol{y}, \mathbf{X}_0)$ pairs for training and 4000 for testing. All convolutional layers in the proximal network have a kernel size of 3 with stride 1 and 1 zero padding. The proposed RNN is trained using Adam optimization [42] with an initial 0.01 learning rate. The network is trained with batch size 32 for 200 epochs. During training, we halve the learning rate if the loss function value, $||\text{vec}(\mathbf{X}) - \text{vec}(\mathbf{X}_0)||_2$, does not decrease for 3 consecutive epochs.

A. Unfolding Different Numbers of Iterations

Normally, proximal gradient descent will stop when a preset convergence criterion is satisfied. For the proposed RNN, we fix in advance the number of iterations for unfolding. To examine the effect of the number of unfolding iterations, we train our RNN with different numbers of unfolding iterations and record their performance in Table I in terms of the exact support recovery rate, the average recovery error, $||\text{vec}(\mathbf{X}) - \text{vec}(\mathbf{X}_0)||_2$, and the average processing time measured on a

system with an i7-6700 CPU and GTX 1080 GPU. We use RNN-k to denote the network constructed from k unfolded iterations of proximal gradient descent. Exact support recovery is achieved when the estimated $\mathbf{X}$ has the same support as the ground-truth $\mathbf{X}_0$. We compare our method to the $\ell_{2,1}$ optimization method proposed in [22].

From Table I we can observe that, when the number of unfolding iterations is ≥ 3, the proposed network achieves a comparable exact support recovery rate compared to the $\ell_{2,1}$ optimization method. However, the proposed network is much more efficient compared to the $\ell_{2,1}$ optimization method solved via CVX [43].

B. Effect of $\mathbf{\Phi}$

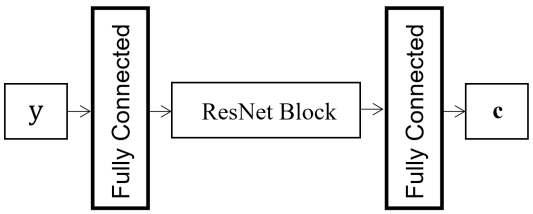
In (5), we see that the proximal operator is independent of $\mathbf{\Phi}$, which implies that the proximal network could also be independent of $\mathbf{\Phi}$. Namely, for a different sensing matrix $\mathbf{\Phi}$, our proposed network can be reused by simply replacing the $\mathbf{\Phi}$ matrix in the recurrent network and plugging in the pre-trained proximal network. To verify that, we construct 4000 $(\boldsymbol{y}, \mathbf{X}_0)$ pairs for different sensing matrices $\mathbf{\Phi}$ following the same random generation process and record the exact support recovery rate of our proposed recurrent network in Table II without retraining the network. From the results we can observe that our proposed network can be easily adaptive to other systems with a different sensing matrix $\mathbf{\Phi}$.

C. Comparison to A Generic ResNet

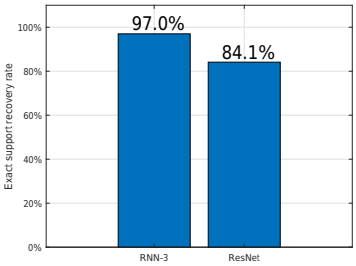
Recovering the indices of non-zero columns in $\mathbf{X}_0$ is equivalent to recovering the indices of non-zero entries in the vector $\boldsymbol{c}$. Thus, an alternative approach for solving the support recovery problem is to directly recover a sparse vector $\boldsymbol{c}$ from the observation $\boldsymbol{y}$ via a generic network without applying the lifting technique to construct $\mathbf{X}$. To examine this approach, we design a generic ResNet [44] shown in Fig. 2(a), in which the ResNet block consists of a sequential stack of three independent ProximalNets whose structures are shown in Fig. 1. The fully connected layers accommodate the data sizes for the input and output accordingly. Because in our proposed recurrent network all ProximalNets share the same weights, the number of learnable weights in RNN-3 is about 75.8% of the number of learnable weights in the compared generic ResNet. We record the exact support recovery rate of the RNN-3 and the generic ResNet in Fig. 2(b), from which we can observe that incorporating the optimization technique into the network design improves the network performance significantly, even with a smaller number of learnable weights.

D. Effect of J

In this section, without retraining the network, we examine the exact support recovery rate of the proposed network RNN-3 when $\mathbf{X}_0$ has different numbers, J, of non-zero columns. The result is shown in Fig. 3, from which we can observe that although RNN-3 is trained only on data with $J = 2$, it is robust to the change of J and achieves a comparable support recovery rate of the $\ell_{2,1}$ optimization method [22].



(a) A generic ResNet.



(b) Exact support recovery rate.

Fig. 2: The comparison between RNN-3 and a generic ResNet that predicts the sparse vector $\boldsymbol{c}$ directly for support recovery. (a) The ResNet for comparison, whose ResNet block consists of a sequential stack of three independent ProximalNets. (b) The exact support recovery rate of RNN-3 and the generic ResNet.

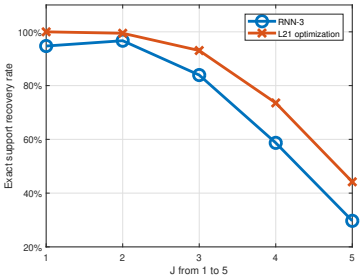


Fig. 3: The exact support recovery rate of the proposed RNN-3 with different numbers J of non-zero columns in $\mathbf{X}_0$.

E. Approximately Column-wise Sparse $\mathbf{X}$

Due to system noise, the matrix $\mathbf{X}$ of interest might be only approximately column-wise sparse. For example, we suppose the system observes $\boldsymbol{y} = \mathbf{\Phi} \cdot \text{vec}(\mathbf{X}) + \boldsymbol{n}$ where the signal of interest $\text{vec}(\mathbf{X}) = \mathbf{G} \cdot \text{vec}(\mathbf{X}_0)$ and $\mathbf{G} = \mathbf{I} + \mathbf{H}$. $\mathbf{X}_0$ is the original column-wise sparse matrix, $\mathbf{I} \in \mathbf{R}^{KM \times KM}$ is the identity matrix, and $\mathbf{H} \in \mathbf{R}^{KM \times KM}$ is a random Gaussian matrix with entries drawn from the Gaussian distribution with mean 0 and standard deviation $\sigma_{\mathbf{G}}$. In this section, we set $\sigma_{\mathbf{G}} = 0.01$ and record the recovery performance of the $\ell_{2,1}$ optimization method [22] and the re-trained RNN-3 network in Table III. We see that even when the matrix of interest is not strictly column-wise sparse, the proposed network can still be very effective in recovering the signal.

TABLE I: The performance of the proposed RNN with different numbers of unfolding iterations. Specifically, RNN-k denotes that we unfold k iterations of proximal gradient descent and thus RNN-k contains k recurrent blocks.

	$\ell_{2,1}$ optimization [22]	RNN-1	RNN-3	RNN-5	RNN-7
Exact support recovery rate	100.0%	75.6%	97.0%	97.2%	98.7%
Average recovery error	4.69	4.35	2.35	1.79	1.16
Average processing time (seconds)	0.58	0.69×10^{-4}	1.61×10^{-4}	2.54×10^{-4}	3.51×10^{-4}

TABLE II: The exact support recovery rate of the proposed RNN-3 with different sensing matrices $\mathbf{\Phi}$.

	Φ_1	Φ_2	Φ_3
The exact support recovery rate	97.0%	97.6%	96.8%

TABLE III: The average recovery error with the approximately column-wise sparse $\mathbf{X}$

	$\ell_{2,1}$ optimization [22]	RNN-3
Average recovery error	5.40	3.67

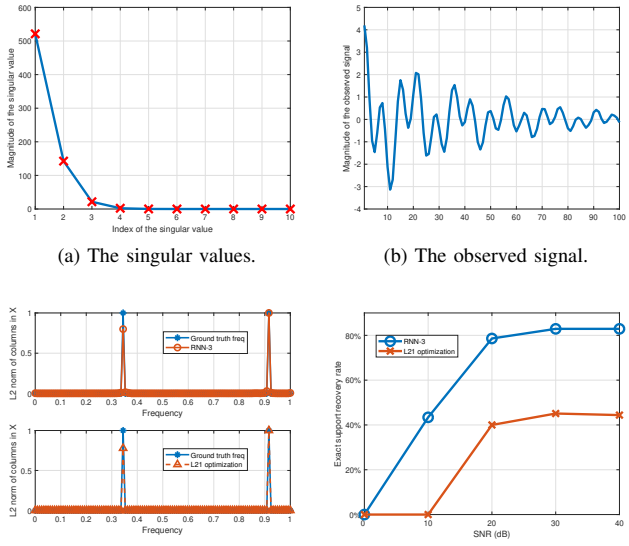
F. Frequency Estimation for Damped Sinusoids

In this section, we apply the proposed RNN-3 to the frequency estimation problem for damped sinusoids introduced in Section I-C. Specifically, each column in $\mathbf{A}$ has the form $\boldsymbol{a}(\theta) = [1, \cos(\theta), \cdots, \cos(\theta(N-1))]^T$ where we choose θ from among $M = 120$ evenly separated values in $[0, 1]$. $\mathbf{D}_j$ contains the damping signal $e^{-\alpha \cdot n}$ for $n = \{0, 1, \cdots, N-1\}$ in its diagonal entries and we sample α uniformly in $[0, 0.05]$ such that a maximally damped sinusoid ($\alpha = 0.05$) would see its amplitude decay by around $100\times$ between its first and last samples when $N = 100$. To approximate the damping signal, we generate 10000 damping signals whose damping coefficients range from 0 to 0.05 and perform the singular value decomposition (SVD) to construct the subspace matrix $\mathbf{B}$. The magnitudes of the singular values of $\mathbf{B}$ are shown in Fig. 4(a), based on which we set $K = 3$. For each observed signal, the $J = 2$ frequencies of damped sinusoids are uniformly selected on the frequency grids with at least $2/M$ separation and signal magnitudes are uniformly sampled in $[0.5, 1.0]$. The observation is also contaminated with additive Gaussian noise $\tilde{\boldsymbol{n}}$ as in (4). An observed signal with 30 dB signal-to-noise ratio (SNR),

$$SNR(dB) = 20\log_{10}\left(\frac{||\sum_{j=1}^{M} c_j \mathbf{D}_j \boldsymbol{a}(\theta_j)||_2}{||\tilde{\boldsymbol{n}}||_2}\right),$$

is shown in Fig. 4(b) and whose ground truth frequencies are indicated in Fig. 4(c) using magnitude-1 spikes.

Since we constructed $\boldsymbol{y}$ via the $\mathbf{D}_j$ matrices directly, for network training, we construct the target $\mathbf{X} = [c_1\boldsymbol{h}_1 \quad c_2\boldsymbol{h}_2 \quad \cdots \quad c_M\boldsymbol{h}_M]$ via the sampled c_j and $\boldsymbol{h}_j$ estimated from $\min_{\boldsymbol{h}_j} ||\mathbf{D}_j - \text{diag}(\mathbf{B}\boldsymbol{h}_j)||_F^2$. The sizes of the training, validation, and testing datasets are 16000, 4000, and 1000 respectively. The ℓ_2 norm of the columns in the recovered $\mathbf{X}$ for the signal observed in Fig. 4(b) are shown in Fig. 4(c) for both the RNN-3 and the $\ell_{2,1}$ optimization method [22]. The exact support recovery rate on the testing dataset with different SNRs is shown in Fig. 4(d). From the results we can observe that when $\mathbf{\Phi}$ is only an approximation of the sensing process, the proposed RNN-3 is still very robust in estimating the support.



(a) The singular values. (b) The observed signal.

(c) ℓ_2 norm of columns in recovered $\mathbf{X}$. (d) Exact support recovery rate.

Fig. 4: Frequency estimation for damped sinusoids. (a) The singular values of the collected damped signals' matrix. (b) One observed signal with 30 dB SNR. (c) The ground-truth frequencies and the ℓ_2 norm of the columns in the recovered $\mathbf{X}$. (d) The exact support recovery rate with different SNRs.

IV. Conclusion

In this paper, we study the support recovery problem of sparse signals with non-stationary modulation via a proximal gradient descent inspired data-driven method. With the common modulating signal subspace assumption and using the lifting technique, we reformulate the support recovery problem into a column-wise sparse matrix recovery problem, which can be effectively solved via the $\ell_{2,1}$ norm regularized quadratic minimization. By unfolding the proximal gradient descent for the $\ell_{2,1}$ norm regularized quadratic minimization, we propose a novel recurrent neural network to solve the original support recovery problem. Simulation results show that the proposed network is extremely efficient, can be adaptive to different sensing matrices without retraining the network, and can be applied to the cases where the matrix of interest is not strictly column-wise sparse and where we only know an approximation of the sensing process.

REFERENCES

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

[2] S. Li, Y. Xie, Q. Li, and G. Tang, "Cubic regularization for differentiable games," in *Smooth Games Optimization and Machine Learning Workshop (NeurIPS 2019)*, Vancouver, Canada, 2019.

[3] S. Foucart and H. Rauhut, *A mathematical introduction to compressive sensing*. Birkhäuser Basel, 2013.

[4] G. Tang, B. N. Bhaskar, P. Shah, and B. Recht, "Compressed sensing off the grid," *IEEE Transactions on Information Theory*, vol. 59, no. 11, pp. 7465–7490, 2013.

[5] Y. Xie, S. Li, G. Tang, and M. B. Wakin, "Radar signal demixing via convex optimization," in *2017 22nd International Conference on Digital Signal Processing (DSP)*, pp. 1–5, IEEE, 2017.

[6] B. Vandereycken, "Low-rank matrix completion by riemannian optimization," *SIAM Journal on Optimization*, vol. 23, no. 2, pp. 1214–1236, 2013.

[7] Y. Xie, D. Liu, H. Mansour, and P. T. Boufounos, "Robust parameter estimation of contaminated damped exponentials," in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5500–5504, IEEE, 2020.

[8] M. J. Wainwright, "Sharp thresholds for high-dimensional and noisy sparsity recovery using ℓ_1-constrained quadratic programming (lasso)," *IEEE Transactions on Information Theory*, vol. 55, no. 5, pp. 2183–2202, 2009.

[9] S. Ling and T. Strohmer, "Self-calibration and biconvex compressive sensing," *Inverse Problems*, vol. 31, no. 11, p. 115002, 2015.

[10] Y. Xie, Y. Tang, G. Tang, and W. Hoff, "Learning to find good correspondences of multiple objects," in *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 2779–2786, IEEE, 2021.

[11] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 779–788, 2016.

[12] Y. Xie, Z. Wang, W. Pei, and G. Tang, "Fast approximation of nonnegative sparse recovery via deep learning," in *2019 IEEE International Conference on Image Processing (ICIP)*, pp. 2921–2925, IEEE, 2019.

[13] K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," in *International Conference on Machine Learning*, pp. 399–406, Omnipress, 2010.

[14] G. Izacard, S. Mohan, and C. Fernandez-Granda, "Data-driven estimation of sinusoid frequencies," in *Advances in Neural Information Processing Systems*, pp. 5127–5137, 2019.

[15] Y. Xie, G. Tang, and W. Hoff, "Chess piece recognition using oriented chamfer matching with a comparison to cnn," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 2001–2009, IEEE, 2018.

[16] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*, vol. 1. MIT press, 2016.

[17] Y. Xie, M. B. Wakin, and G. Tang, "Data-driven parameter estimation of contaminated damped exponentials," in *2021 55th Asilomar Conference on Signals, Systems, and Computers*, IEEE, 2021.

[18] V. Monga, Y. Li, and Y. C. Eldar, "Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing," *IEEE Signal Processing Magazine*, vol. 38, no. 2, pp. 18–44, 2021.

[19] P. L. Combettes and J.-C. Pesquet, "Proximal splitting methods in signal processing," in *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, pp. 185–212, Springer, 2011.

[20] D. Yang, G. Tang, and M. B. Wakin, "Super-resolution of complex exponentials from modulations with unknown waveforms," *IEEE Transactions on Information Theory*, vol. 62, no. 10, pp. 5809–5830, 2016.

[21] Y. Xie, M. B. Wakin, and G. Tang, "Simultaneous sparse recovery and blind demodulation," *IEEE Transactions on Signal Processing*, vol. 67, no. 19, pp. 5184–5199, 2019.

[22] Y. Xie, M. B. Wakin, and G. Tang, "Support recovery for sparse signals with unknown non-stationary modulation," *IEEE Transactions on Signal Processing*, vol. 68, pp. 1884–1896, 2020.

[23] S. Ling and T. Strohmer, "Blind deconvolution meets blind demixing: Algorithms and performance bounds," *IEEE Transactions on Information Theory*, vol. 63, no. 7, pp. 4497–4520, 2017.

[24] Y. Chi, "Guaranteed blind sparse spikes deconvolution via lifting and convex optimization," *Journal of Selected Topics in Signal Processing*, vol. 10, no. 4, pp. 782–794, 2016.

[25] Y. Xie, M. B. Wakin, and G. Tang, "Sparse recovery and non-stationary blind demodulation," in *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5566–5570, IEEE, 2019.

[26] Y. Xie, M. B. Wakin, and G. Tang, "Support recovery for sparse recovery and non-stationary blind demodulation," in *2019 53rd Asilomar Conference on Signals, Systems, and Computers*, pp. 235–239, IEEE, 2019.

[27] H. Qarib and H. Adeli, "A new adaptive algorithm for automated feature extraction in exponentially damped signals for health monitoring of smart structures," *Smart Materials and Structures*, vol. 24, no. 12, p. 125040, 2015.

[28] M. Kanemaru, M. Tsukima, T. Miyauchi, and K. Hayashi, "Bearing fault detection in induction machine based on stator current spectrum monitoring," *IEEJ Journal of Industry Applications*, vol. 7, no. 3, pp. 282–288, 2018.

[29] J.-F. Cai, X. Qu, W. Xu, and G.-B. Ye, "Robust recovery of complex exponential signals from random gaussian projections via low rank hankel matrix reconstruction," *Applied and Computational Harmonic Analysis*, vol. 41, no. 2, pp. 470–490, 2016.

[30] A. Ahmed, B. Recht, and J. Romberg, "Blind deconvolution using convex programming," *IEEE Transactions on Information Theory*, vol. 60, no. 3, pp. 1711–1732, 2014.

[31] Y. C. Eldar, W. Liao, and S. Tang, "Sensor calibration for off-the-grid spectral estimation," *Applied and Computational Harmonic Analysis*, 2018.

[32] A. Flinth, "Sparse blind deconvolution and demixing through $\ell_{1,2}$-minimization," *Advances in Computational Mathematics*, vol. 44, no. 1, pp. 1–21, 2018.

[33] A. Mousavi and R. G. Baraniuk, "Learning to invert: Signal recovery via deep convolutional networks," in *2017 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 2272–2276, IEEE, 2017.

[34] Y. Tang, K. Kojima, T. Koike-Akino, Y. Wang, P. Wu, Y. Xie, M. H. Tahersima, D. K. Jha, K. Parsons, and M. Qi, "Generative deep learning model for inverse design of integrated nanophotonic devices," *Laser & Photonics Reviews*, vol. 14, no. 12, p. 2000287, 2020.

[35] G. Izacard, B. Bernstein, and C. Fernandez-Granda, "A learning-based framework for line-spectra super-resolution," in *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3632–3636, IEEE, 2019.

[36] Y. Xie, M. B. Wakin, and G. Tang, "Contaminated multiband signal identification via deep learning," in *2021 IEEE Statistical Signal Processing Workshop (SSP)*, IEEE, 2021.

[37] P. Sprechmann, A. M. Bronstein, and G. Sapiro, "Supervised non-euclidean sparse nmf via bilevel optimization with applications to speech enhancement," in *2014 4th Joint Workshop on Hands-free Speech Communication and Microphone Arrays (HSCMA)*, pp. 11–15, IEEE, 2014.

[38] J. R. Hershey, J. L. Roux, and F. Weninger, "Deep unfolding: Model-based inspiration of novel deep architectures," *arXiv preprint arXiv:1409.2574*, 2014.

[39] Y. Chen and T. Pock, "Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1256–1272, 2016.

[40] R. Liu, S. Cheng, L. Ma, X. Fan, and Z. Luo, "Deep proximal unrolling: Algorithmic framework, convergence analysis and applications," *IEEE Transactions on Image Processing*, vol. 28, no. 10, pp. 5013–5026, 2019.

[41] M. Mardani, Q. Sun, S. Vasawanala, V. Papyan, H. Monajemi, J. Pauly, and D. Donoho, "Neural proximal gradient descent for compressive imaging," in *Advances in Neural Information Processing Systems*, pp. 9596–9606, 2018.

[42] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[43] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1." http://cvxr.com/cvx, Mar. 2014.

[44] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, IEEE, 2016.