Circuits, Systems, and Signal Processing (2019) 38:3616–3643
https://doi.org/10.1007/s00034-019-01058-5

Recurrent Neural Network-Based Dictionary Learning for Compressive Speech Sensing

Yunyun Ji[1,2] · Wei-Ping Zhu[2] · Benoit Champagne[3]

Received: 6 September 2018 / Revised: 23 January 2019 / Accepted: 7 February 2019 /
Published online: 14 February 2019

Abstract

We propose a novel dictionary learning technique for compressive sensing of speech signals based on the recurrent neural network. First, we exploit the recurrent neural network to solve an ℓ_0-norm optimization problem based on a sequential linear prediction model for estimating the linear prediction coefficients for voiced and unvoiced speech, respectively. Then, the extracted linear prediction coefficient vectors are clustered through an improved Linde–Buzo–Gray algorithm to generate codebooks for voiced and unvoiced speech, respectively. A dictionary is then constructed for each type of speech by concatenating a union of structured matrices derived from the column vectors in the corresponding codebook. Next, a decision module is designed to determine the appropriate dictionary for the recovery algorithm in the compressive sensing system. Finally, based on the sequential linear prediction model and the proposed dictionary, a sequential recovery algorithm is proposed to further improve the quality of the reconstructed speech. Experimental results show that when compared to the selected state-of-the-art approaches, our proposed method can achieve superior performance in terms of several objective measures including segmental signal-to-noise ratio, perceptual evaluation of speech quality and short-time objective intelligibility under both noise-free and noise-aware conditions.

Keywords Recurrent neural network · Linear prediction coefficient · Clustering · Sequential recovery algorithm · Compressive sensing

✉ Yunyun Ji
jyy@ece.concordia.ca

Wei-Ping Zhu
weiping@ece.concordia.ca

Benoit Champagne
benoit.champagne@mcgill.ca

[1] School of Electronics and Information, Nantong University, Nantong, China

[2] Department of Electrical and Computer Engineering, Concordia University, Montreal, Canada

[3] Department of Electrical and Computer Engineering, McGill University, Montreal, Canada

1 Introduction

In the last decade, compressive sensing (CS) [5,11], as an alternative for sampling and compression, has achieved great development with its widespread applications in speech processing, image processing, radar and wireless communication. As opposed to conventional Nyquist sampling which firstly performs sampling at a high frequency exceeding twice of the signal bandwidth and then realizes compression, CS can achieve sampling and compression simultaneously, which is beneficial for hardware design and storage resources. Compared to other application areas and in spite of efforts devoted by many researchers, compressive speech sensing still remains in a preliminary stage in terms of both theory and practice. The majority of related works focus on how to apply the CS technique to speech processing tasks such as enhancement [34,35], encoding [14]. In [34], the CS method is applied to achieve speech and noise separation in time–frequency domain. In [35], the compressive sensing matching pursuit (CoSaMP) algorithm [22] is adopted for time-domain speech denoising, which is regarded as an alternative to traditional speech enhancement. A speech coding approach based on sparse linear predictor is proposed in [14] to establish a new speech encoding framework which can improve the coding performance of the traditional linear prediction systems.

Notwithstanding these remarkable advances, the crux of compressive speech sensing still lies on improving the sparsity of speech signals and designing speech-specific recovery algorithms, which is also the basis for the above-mentioned applications but has only achieved limited development to date. One effective method for improving signal sparsity is to employ dictionary learning techniques to train data-driven dictionaries for sparse representation. Typical dictionary learning methods include the method of optimal directions (MOD) [12], the KSVD algorithm [2] and the online dictionary learning (ODL) method [21]. These approaches exploit sparsity-promoting term (ℓ_0 or ℓ_1 norm) within regularized optimization problems to learn overcomplete dictionaries based on a large-scale training dataset. Recently, a dictionary learning method based on principal component analysis (PCA) was proposed for speech unit classification [25]. However, without considering specific characteristics of target signals, these methods cannot well capture the internal structure of signals and thus fail to improve the sparsity of signals with respect to the trained dictionaries.

In this paper, we exploit a sequential linear prediction model in conjunction with a recurrent neural network (RNN) [16]-based optimization algorithm to construct the dictionary with a predetermined structure for speech signals. Moreover, in view of the sequential linear prediction model, we incorporate the intra-frame correlation of speech signals into the existing sparse recovery algorithms. In other words, this sequential recovery algorithm is proposed to leverage the latent information from the previously reconstructed frame and improve the quality of reconstructed speech.

Based on these proposed techniques, this paper presents a new compressive speech sensing system which handles the voiced and unvoiced speech separately. The proposed system is divided into two parts: the training stage and the application stage. In the training stage, a large number of training data are utilized to learn a dictionary for voiced and unvoiced speech, respectively, with our proposed dictionary learning method. In the application stage, we employ the sequential recovery algorithm to realize effective reconstruction of speech with the trained dictionaries from the training stage. Moreover, we devise a decision module to select an appropriate dictionary for the sequential recovery algorithm.

This paper is arranged as follows. In Sect. 2, we give a brief description of CS, dictionary learning and RNN for optimization. In Sect. 3, we concretely describe our proposed compressive speech sensing system and explain the principles of the modules involved. Section 4 evaluates the performance of our proposed system with three objective measures and compare the results with four reference methods. In Sect. 5, we conclude the paper with a brief summary of our contributions.

2 Background

In this section, we present an overview of CS, dictionary learning and RNN for optimization.

2.1 Compressive Sensing

CS provides an effective sampling and compression framework for sparse or compressible (approximately sparse) signals and is able to realize high-quality reconstruction of these signals with fewer measurements, providing a potential means for natural signal processing such as images and speech signals in different kinds of applications. Three key factors are involved in the CS theory: sparse representation, sensing matrix and sparse recovery [10]. Sparse representation is the precondition of CS and usually employs the dictionary learning method to learn an overcomplete dictionary to capture the internal structure of signals [39], which is described in Sect. 2.2. In this subsection, we focus on the sensing matrix and sparse recovery algorithm.

Given an S-sparse data vector $\boldsymbol{x} \in \mathbb{R}^N$ (i.e., this vector has S nonzero elements), a sensing matrix $\boldsymbol{\Phi} \in \mathbb{R}^{M \times N}$ ($M \leq N$) can be designed to simultaneously sample and compress the signal and generate a measurement vector as

$$\boldsymbol{y} = \boldsymbol{\Phi x}. \tag{1}$$

Ideally, the projection of $\boldsymbol{x}$ into a lower-dimensional space through the matrix $\boldsymbol{\Phi}$ entails no loss of information so that it remains possible to reconstruct the original sparse vector $\boldsymbol{x}$ from its associated measurement vector $\boldsymbol{y}$. To this end, the sensing matrix above, as a linear operator for CS, is designed to satisfy the restricted isometry property (RIP) [6], i.e., the condition

$$(1 - \delta_S) \left\| \boldsymbol{x} \right\|_2^2 \leq \left\| \boldsymbol{\Phi x} \right\|_2^2 \leq (1 + \delta_S) \left\| \boldsymbol{x} \right\|_2^2 \tag{2}$$

should hold for all S-sparse signals with $\delta_S \in (0, 1)$.

When considered as linear projection operators, some random matrices, including Gaussian, partial Fourier and Bernoulli random matrices, have been shown to conform to the RIP [7]. The advantages of random matrices are twofold, namely, the universality and democracy [19].

Another key feature of CS is the possibility to recover the sparse signal $\boldsymbol{x}$ from the measurement vector $\boldsymbol{y}$. In practice, it is difficult to find out the true solution from the innumerable solutions of (1) without any constraints. However, the desirable sparsity can guarantee accurate reconstruction of $\boldsymbol{x}$ from the following optimization problem [8],

$$\min \|\boldsymbol{x}\|_0 \qquad \text{s.t.} \quad \boldsymbol{y} = \boldsymbol{\Phi x} \tag{3}$$

where the ℓ_0 norm counts the nonzeros in $\boldsymbol{x}$.

It has been proved that as long as the sensing matrix satisfies RIP with the restricted isometry constant $\delta_{2S} \in (0, 1)$, the optimization technique in (3) can recover all S-sparse signals. However, as the above ℓ_0-norm minimization problem is NP-hard, the feasible alternatives [13] for (3) come into two flavors: greedy pursuit algorithms such as orthogonal matching pursuit (OMP) [32] and convex relaxation methods such as basis pursuit (BP) [9] which replaces the ℓ_0 norm with the convex ℓ_1 norm. In the BP algorithm, the above optimization problem in (3) can be transformed into

$$\min \|\boldsymbol{x}\|_1 \qquad \text{s.t.} \quad \boldsymbol{y} = \boldsymbol{\Phi x}. \tag{4}$$

2.2 Dictionary Learning

The goal of dictionary learning is to extract significant information and reduce dimensionality [4]. A well-trained dictionary can provide compact representation for specific categories of signals. This kind of representation is named sparse representation, meaning that signals can be expressed as a linear combination of a portion of atoms, i.e., the column vectors, in the redundant dictionary. The number of the selected atoms should be much smaller than the signal dimension, which is regarded as the sparsity level of the signal with respect to the specified dictionary [31]. In the context of CS, it is well acknowledged that sparser representation can lead to higher compression rate at the same level of reconstruction quality [20]. Thus, an effective dictionary learning approach can improve the performance of the CS system.

One of the most typical dictionary learning algorithms is the KSVD algorithm [2] which can learn a dictionary and a sparse coefficient matrix simultaneously by solving the following optimization problem.

$$\min_{\boldsymbol{\Psi},\boldsymbol{\Theta}} \|\boldsymbol{X} - \boldsymbol{\Psi\Theta}\|_F^2 \quad \text{s.t.} \quad \|\boldsymbol{\theta}_i\|_0 \le S \qquad \text{for} \quad i = 1, 2, \ldots, L \tag{5}$$

where $\boldsymbol{X} = \left[\boldsymbol{x}_1\ \boldsymbol{x}_2 \cdots \boldsymbol{x}_L\right]$ denotes the data matrix with the signal vectors as its columns, $\boldsymbol{\Theta} = \left[\boldsymbol{\theta}_1\ \boldsymbol{\theta}_2 \cdots \boldsymbol{\theta}_L\right]$ denotes the sparse coefficient matrix, the columns of which are the sparse coefficient vectors of the corresponding signal vectors in $\boldsymbol{X}$ with respect to the dictionary $\boldsymbol{\Psi}$, S is the sparsity level of the signal, and $\|\cdot\|_F$ denotes the Frobenius norm of a matrix (i.e., referring to the square root of the sum of the squares of all the elements in the matrix). The KSVD algorithm updates the dictionary $\boldsymbol{\Psi}$ and the sparse coefficient matrix $\boldsymbol{\Theta}$ alternately in each iteration. In detail, the KSVD algorithm employs the OMP algorithm at the sparse coding stage to estimate the sparse coefficient matrix $\boldsymbol{\Theta}$; the singular value decomposition (SVD) is then utilized in the codebook update stage to update all the atoms of the dictionary. In [2], the KSVD algorithm has been able to train an effective dictionary for image inpainting and compression. Different from the KSVD algorithm, the MOD [12] updates the dictionary iteratively by exploiting the gradient of the loss function in (5) with respect to $\boldsymbol{\Psi}$ with fixed $\boldsymbol{\Theta}$, i.e., $\boldsymbol{\Psi} = \boldsymbol{X}\boldsymbol{\Theta}^T \left(\boldsymbol{\Theta}\boldsymbol{\Theta}^{\mathrm{T}}\right)^{-1}$.

The ODL algorithm [21] was proposed to learn the dictionary based on large-scale training data, aiming at reducing memory cost and computational complexity. In contrast with the KSVD and the MOD, a single signal or a mini-batch, instead of the whole training set, is processed at each iteration of the ODL method. The following optimization problem was addressed to estimate the sparse coefficient matrix in the ODL algorithm [33]:

$$\boldsymbol{\Theta} = \arg\min_{\boldsymbol{\Theta}} \frac{1}{2} \|\boldsymbol{X} - \boldsymbol{\Psi}\boldsymbol{\Theta}\|_F^2 + \lambda \|\boldsymbol{\Theta}\|_1 . \tag{6}$$

Then in the dictionary updating step, with the estimated sparse coefficient matrix $\boldsymbol{\Theta}$ from the sparse coding step, the dictionary $\boldsymbol{\Psi}$ is updated by solving

$$\boldsymbol{\Psi} = \arg\min_{\boldsymbol{\Psi}} -2\mathrm{tr}(\boldsymbol{X}\boldsymbol{\Theta}^T\boldsymbol{\Psi}^T) + \mathrm{tr}(\boldsymbol{\Theta}\boldsymbol{\Theta}^T\boldsymbol{\Psi}^T\boldsymbol{\Psi}). \tag{7}$$

In the PCA-based dictionary learning method [25], speech signals are firstly grouped into several clusters via the K-means algorithm [1]. Then, PCA is applied to the clustered speech signals to obtain the dictionary. As PCA provides discriminative information, this method can achieve good performance in speech units classification.

2.3 Recurrent Neural Network for Optimization

Neural networks have been ingeniously applied to solve various optimization problem for more than 3 decades [30]. In particular, various neural network models have been developed to solve constrained convex optimization problems, the key point of which is to leverage neurodynamic systems with state vectors to approximate desirable solutions [17]. In [36–38], a series of algorithms were proposed based on the RNN, a prevalent neural network model, to address different convex optimization problems for autoregressive parameter estimation under different environments. Specifically, an algorithm named noise-constrained least squares was developed in [37] to solve the quadratic optimization problem with linear inequality constraints for prediction coefficients estimation under Gaussian noise environment and then applied to speech enhancement in [38]. Meanwhile, an ℓ_1-norm minimization problem was solved by a generalized least absolute deviation algorithm in [36] for corresponding coefficients estimation in the presence of non-Gaussian noise. These algorithms take advantage of the learning ability of the RNN to approximate optimal solutions of optimization problems, which have low computational complexity and can be implemented in real-time systems.

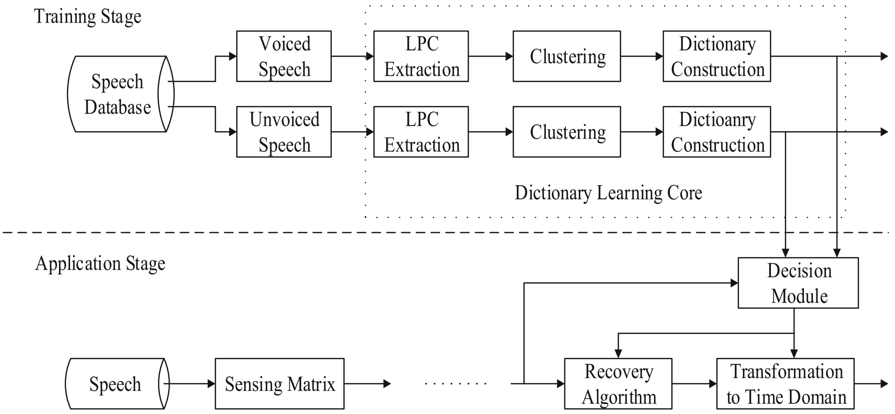


Fig. 1 Block diagram for our proposed compressive speech sensing system

3 Proposed Compressive Speech Sensing System

3.1 System Overview

We illustrate the block diagram of our proposed compressive speech sensing system in Fig. 1. The whole system is divided into two stages: the training stage and the application stage. In the training stage, a vast number of speech signals are employed to train the data-driven dictionary. The training speech is categorized into two groups, i.e., voiced speech and unvoiced speech. Unvoiced speech acts like white noise, resulting in unsatisfactory sparsity with respect to most existing orthonormal bases and overcomplete dictionaries. In most of the literature on compressive speech sensing, unvoiced speech is given less importance and not processed in a separate way [28]. However, considering the structural differences between voiced and unvoiced speech [14], it would seem that the overall reconstruction performance can be improved if a specific dictionary is learned for unvoiced speech. This observation motivates us to construct dictionaries for the voiced and unvoiced speech separately. In other words, we incorporate the voiced and unvoiced labels to the speech frames used during the course of dictionary learning, which is anticipated to improve the overall sparsity of speech and further enhance sparse reconstruction performance of CS. The proposed dictionary learning method in the training stage is divided into four steps as follows:

Step 1: Classify the training speech data into two groups: voiced speech and unvoiced speech.

Step 2: Extract linear prediction coefficients (LPCs) of voiced and unvoiced speech based on the sequential linear prediction model by using the RNN sparse excitation LPC algorithm, which will be developed in Sect. 3.2.

Step 3: Apply the clustering algorithm to the LPCs of voiced and unvoiced speech and obtain cluster centroids needed to construct the voiced and unvoiced codebooks, respectively.

Step 4: Using a predetermined structure, fill the column vectors of the two codebooks into a union of structured matrices, respectively, to form the voiced and unvoiced dictionary, denoted as $\boldsymbol{\Psi}^{\mathrm{v}}$ and $\boldsymbol{\Psi}^{\mathrm{u}}$, respectively.

In contrast with the traditional dictionary learning methods [2,12,33], our proposed method takes advantage of the speech features, the speech generative model, the neural network-based optimization technique and the clustering algorithm to capture the inherent structure of speech and construct effective dictionaries.

In the application stage, the measurement vector $\boldsymbol{y}$ is obtained by application of sensing matrix $\boldsymbol{\Phi}$ to the speech signal vector $\boldsymbol{x}$ and then "transmitted" through a given channel to the receiver. Due to the presence of noise or other imperfection in the transmission channel, the received measurement vector might be different from $\boldsymbol{y}$. Accordingly, at the receiver, the sensing matrix, the trained dictionary and the received measurement vector, are input to the recovery algorithm module to estimate the sparse coefficients of speech. Finally, we can reconstruct the speech signal by multiplying the dictionary and the estimated sparse coefficient vector. However, at the receiver, we cannot directly decide the type of the speech frames (i.e., voiced or unvoiced) from the received measurement vectors because the permutations induced by the sensing matrix makes the observations behave like Gaussian noise. To address this issue, a decision module is designed to help us select the desired dictionary, i.e., for either voiced or unvoiced speech. Specifically, in the decision module, we can employ the residual error of the measurements as the metric for selecting either the voiced or unvoiced dictionary for reconstruction, owing to the fact,

$$(1-\delta_{2S})\left\|\boldsymbol{\theta}-\hat{\boldsymbol{\theta}}\right\|_2^2 \leq \left\|\boldsymbol{y}-\boldsymbol{\Phi}\boldsymbol{\Psi}\hat{\boldsymbol{\theta}}\right\|_2^2 \leq (1+\delta_{2S})\left\|\boldsymbol{\theta}-\hat{\boldsymbol{\theta}}\right\|_2^2$$

where $\boldsymbol{\theta}$ is the true sparse coefficient vector of the speech signal $\boldsymbol{x}$ with respect to the dictionary $\boldsymbol{\Psi}$ and $\hat{\boldsymbol{\theta}}$ is the estimated sparse coefficient vector from the recovery algorithm.

Decision module

Step 1: Use the recovery algorithm to estimate the sparse coefficient vector with respect to the dictionary $\boldsymbol{\Psi}^{\mathrm{v}}$, denoted as $\hat{\boldsymbol{\theta}}_{\mathrm{I}}$. According to RIP, the residual error can be measured as $r_{\mathrm{I}} = \left\|\boldsymbol{y}-\boldsymbol{\Phi}\boldsymbol{\Psi}^{\mathrm{v}}\hat{\boldsymbol{\theta}}_{\mathrm{I}}\right\|_2^2$

Step 2: Use the recovery algorithm to estimate the sparse coefficient vector with respect to the dictionary $\boldsymbol{\Psi}^{\mathrm{u}}$, denoted as $\hat{\boldsymbol{\theta}}_{\mathrm{II}}$. The residual error can be measured as $r_{\mathrm{I}} = \left\|\boldsymbol{y}-\boldsymbol{\Phi}\boldsymbol{\Psi}^{\mathrm{u}}\hat{\boldsymbol{\theta}}_{\mathrm{II}}\right\|_2^2$

Step 3: If $r_{\mathrm{I}} < r_{\mathrm{II}}$, the dictionary $\boldsymbol{\Psi}^{\mathrm{v}}$ will be employed as the sparsifying matrix; otherwise, the dictionary $\boldsymbol{\Psi}^{\mathrm{u}}$ will be selected

3.2 New LPC Extraction Algorithm

In this subsection, we will investigate the dictionary learning core used in the training stage in Fig. 1 to construct fine-tuning dictionaries for the speech signals. The dictionary learning core is composed of three modules including LPC extraction, clustering and dictionary construction. In what follows, a new LPC extraction algorithm for dictionary learning is first proposed.

RNN-based LPC Extraction Algorithm As we know, the regular P-order linear predictor can approximate a sample of the ith frame $x_i\,(n)$ as

$$x_i\,(n) = \sum_{p=1}^{P} a_i(p)x_i\,(n-p) + e_i\,(n) \tag{8}$$

where $\{a_i(p)\}$ are the prediction coefficients and $e_i\,(n)$ is the prediction error. However, it is conspicuous that the first P samples in this frame, i.e., $x_i(1), x_i(2), \ldots, x_i(P)$, cannot be fully estimated, resulting in an unreasonable prediction error. In this case, we propose to take advantage of the intra-frame and the inter-frame coherence to address this problem. The samples of the $(i-1)$ th frame $\{x_{i-1}(N), x_{i-1}(N-1), \ldots, x_{i-1}(N+1-P)\}$ are utilized to take place of the samples $\{x_i(0), x_i(-1), \ldots, x_i(1-P)\}$ in (8). Subsequently, the linear prediction model in (8) can be changed to

$$x_i(n) = \begin{cases} \sum_{p=1}^{P} a_i(p)x_{i-1}(N+1-p)+e_i(1), & n=1; \\ \sum_{p=1}^{n-1} a_i(p)x_i(n-p)+\sum_{p=n}^{P} a_i(p)x_{i-1}(N+n-p)+e_i(n), & 1<n\le P; \\ \sum_{p=1}^{P} a_i(p)x_i(n-p)+e_i(n), & P+1\le n\le N. \end{cases} \tag{9}$$

This new linear prediction model in (9) is named as the sequential linear prediction model in this paper. Based on this model, we propose the following optimization technique to estimate the prediction coefficient vector, i.e.,

$$\boldsymbol{a}_i = \arg\min_{\boldsymbol{a}_i} \frac{1}{2}\,\|\boldsymbol{x}_i - \boldsymbol{D}\boldsymbol{a}_i - \boldsymbol{e}_i\|_2^2 \qquad \text{s.t.} \quad \|\boldsymbol{e}_i\|_0 \le S \tag{10}$$

where $\boldsymbol{x}_i = \left[x_i\,(1)\; x_i\,(2)\; \cdots\; x_i\,(N)\right]^{\mathrm{T}}$, $\boldsymbol{e}_i = \left[e_i(1)\; e_i(2)\; \cdots\; e_i(N)\right]^{\mathrm{T}}$, and

$$\boldsymbol{D} = \begin{bmatrix} x_{i-1}\,(N) & x_{i-1}\,(N-1) & x_{i-1}\,(N-2) & \cdots & x_{i-1}\,(N+1-P) \\ x_i\,(1) & x_{i-1}\,(N) & x_{i-1}\,(N-1) & \cdots & x_{i-1}\,(N+2-P) \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ x_i(P-1) & x_i(P-2) & \cdots & x_i(1) & x_{i-1}(N) \\ x_i(P) & x_i(P-1) & x_i(P-2) & \cdots & x_i(1) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_i(N-1) & x_i(N-2) & x_i(N-3) & \cdots & x_i(N-P) \end{bmatrix}.$$

The proposed optimization technique in (10) can estimate the prediction coefficient vector with a guaranteed sparsity of the prediction error vector, which is different from the traditional autocorrelation method [27]. Moreover, as the prediction error vector $\boldsymbol{e}_i$ is also the sparse coefficient vector of the speech signal $\boldsymbol{x}_i$ with respect to our proposed dictionary in Sect. 3.3, the residual $\boldsymbol{e}_i$ is regarded as a part of the approximation for the speech signal.

In [37], the RNN was used to solve the ℓ_2-norm minimization problem by converting it into a dynamic system which can yield the optimal solution through tracking its state trajectory. Thus, the optimization problem in (10) can also be solved with the RNN in a greedy way. The pseudo-code of our proposed RNN sparse excitation LPC (RSEL) algorithm is summarized as follows. For simplicity, we omit the frame index of the prediction coefficient vector and the prediction error vector.

RNN Sparse Excitation LPC (RSEL) Algorithm

```
Input: $\boldsymbol{x}_i$, $\boldsymbol{D}$, sparsity level $S$, step size $\beta$, stopping criterion $\eta$
Initialization: $\boldsymbol{a}^0 = \boldsymbol{0}$, $\boldsymbol{e}^0 = \boldsymbol{0}$, $\mu = \|\boldsymbol{D}\|_F^2$, $\boldsymbol{x}_i = \boldsymbol{x}_i/\mu$, $\boldsymbol{D} = \boldsymbol{D}/\mu$
Iteration: at the $k$th iteration,
1: $\widetilde{\Omega}^k$ ={$2S$ indices of the largest magnitude entries in the vector $\boldsymbol{D}\boldsymbol{a}^{k-1} - \boldsymbol{x}_i$};
2: $\widetilde{\boldsymbol{e}}^k = (1-\beta)\boldsymbol{e}^{k-1} + \beta(\boldsymbol{D}\boldsymbol{a}^{k-1} - \boldsymbol{x}_i)_{\widetilde{\Omega}^k}$;
3: $\Omega^k$ ={$S$ indices of the largest magnitude entries in the vector $\widetilde{\boldsymbol{e}}^k$} ;
4: $\boldsymbol{e}^k = \widetilde{\boldsymbol{e}}^k_{\Omega^k}$;
5: $\boldsymbol{a}^k = (1 - \beta\boldsymbol{D}^{\mathrm{T}}\boldsymbol{D})\boldsymbol{a}^{k-1} + \beta(\boldsymbol{D}^{\mathrm{T}}\boldsymbol{e}^{k-1} + \boldsymbol{D}^{\mathrm{T}}\boldsymbol{x}_i)$;
6: If $\left\|\boldsymbol{a}^k - \boldsymbol{a}^{k-1}\right\|_2 \le \eta$, quit the iteration. The last iteration index is denoted as $K$.
Output: The prediction coefficient vector $\boldsymbol{a}_{\mathrm{RSEL}} = \boldsymbol{a}^K$; the prediction error vector $\boldsymbol{e}_{\mathrm{RSEL}} = \boldsymbol{e}^K$.
```

At each iteration, the approximation residual of $\boldsymbol{x}_i$ calculated from the estimate of the prediction coefficient vector $\boldsymbol{a}^{k-1}$ in the previous iteration can be regarded as a proxy for the prediction error vector. The $2S$ largest magnitude components in this residual vector are located, and the corresponding indices are denoted as a set $\widetilde{\Omega}^k$. With this preliminary support, the intermediate estimate $\widetilde{\boldsymbol{e}}^k$ can work as an approximation for the prediction error vector with an extended support. It is obvious that the support (namely indices of the nonzero entries) of $\widetilde{\boldsymbol{e}}^k$ can be represented as $\mathrm{supp}(\widetilde{\boldsymbol{e}}^k) = \mathrm{supp}(\boldsymbol{e}^k) \bigcup \widetilde{\Omega}^k$, the cardinality of which should be up to $3S$. With the predetermined sparsity constraint, we need to prune this support by retaining the S largest magnitude entries in $\widetilde{\boldsymbol{e}}^k$ to generate the estimate $\boldsymbol{e}^k$. The last step of the iteration is to update the estimate of the prediction coefficient vector based on the interaction between the prediction coefficient vector and the prediction error vector. In Table 1, we provide the computational complexity of the steps at each iteration, based on the standard matrix-vector multiplication. Thus, each iteration of our proposed algorithm is completed in at most $O(NP)$ operations.

Mean Squared Error Analysis In the following Theorem 1, we upper bound the mean squared error of $\boldsymbol{a}_{\mathrm{RSEL}}$ to show the superiority of the proposed RSEL algo-

Table 1 Computational complexity of each step in the RSEL algorithm

Step	1	2	3	4	5	Total
Computational complexity	$O(NP)$	$O(N)$	$O(N)$	$O(N)$	$O(NP)$	$O(NP)$

rithm to the traditional autocorrelation method [27]. In the theorem statement, the result of the autocorrelation method, denoted as $\boldsymbol{a}_{\mathrm{LS}}$, corresponds to the solution of $\min \frac{1}{2} \|\boldsymbol{x} - \boldsymbol{D}\boldsymbol{a}\|_2^2$, while the desired prediction coefficient vector in the linear prediction model is represented as $\boldsymbol{a}_{\mathrm{d}}$. We also note that for simplicity and convenience, we omit the frame index in the analysis and discussions.

Theorem 1 *Suppose that the matrix $\boldsymbol{D}$ has a full-column rank. If the support Ω of the estimate $\boldsymbol{e}_{\mathrm{RSEL}}$ satisfies $\Omega = \{S$ indices of the largest magnitude entries in $\boldsymbol{x} - \boldsymbol{D}\boldsymbol{a}_{\mathrm{d}}\}$ and $\boldsymbol{e}_{\mathrm{RSEL}} = \boldsymbol{e}_{\Omega}$, then*

$$E\left[\|\boldsymbol{a}_{\mathrm{RSEL}} - \boldsymbol{a}_{\mathrm{d}}\|_2^2\right] < E\left[\|\boldsymbol{a}_{\mathrm{LS}} - \boldsymbol{a}_{\mathrm{d}}\|_2^2\right]. \tag{11}$$

Proof We exploit the method in [37] to prove Theorem 1. The linear prediction model can be written as

$$\boldsymbol{x} - \boldsymbol{D}\boldsymbol{a}_{\mathrm{d}} = \boldsymbol{e} = \boldsymbol{e}_{\Lambda} + \boldsymbol{e}_{\Omega} \tag{12}$$

where $\Lambda = \{1, 2, \ldots, N\}\backslash\Omega$ is the complement of Ω. As the nonzero entries of $\boldsymbol{e}_{\Omega}$ correspond to the S largest entries in the desired prediction error vector $\boldsymbol{e}$, the entries of $\boldsymbol{e}_{\Lambda}$ corresponding to the indices in Λ can be considered as i.i.d. Gaussian random variables with zero mean. Hence, $\boldsymbol{e}_{\Lambda}$ is uncorrelated with $\boldsymbol{e}_{\Omega}$.

As mentioned above, $\boldsymbol{a}_{\mathrm{LS}}$ is the solution of $\frac{1}{2} \|\boldsymbol{x} - \boldsymbol{D}\boldsymbol{a}\|_2^2$ and then according to the Gauss–Markov theorem [24], we have the covariance matrix for this solution as

$$E[(\boldsymbol{a}_{\mathrm{LS}} - \boldsymbol{a}_{\mathrm{d}})(\boldsymbol{a}_{\mathrm{LS}} - \boldsymbol{a}_{\mathrm{d}})^{\mathrm{T}}] = (\boldsymbol{D}^{\mathrm{T}}\boldsymbol{R}_e^{-1}\boldsymbol{D})^{-1} \tag{13}$$

where $\boldsymbol{R}_e = E[\boldsymbol{e}\boldsymbol{e}^{\mathrm{T}}]$.

The vector $\boldsymbol{a}_{\mathrm{RSEL}}$ can minimize the loss function $\frac{1}{2} \|\boldsymbol{x} - \boldsymbol{D}\boldsymbol{a} - \boldsymbol{e}_{\mathrm{RSEL}}\|_2^2$, and the support of $\boldsymbol{e}_{\mathrm{RSEL}}$ is the same as that of $\boldsymbol{e}_{\Omega}$. Let $\boldsymbol{u} = \boldsymbol{e}_{\Omega} - \boldsymbol{e}_{\mathrm{RSEL}}$ represent the estimation error. It can be seen that the support of $\boldsymbol{u}$ is also Ω. Consequently, $\boldsymbol{a}_{\mathrm{RSEL}}$ can be regarded as the LS solution of

$$\boldsymbol{x} - \boldsymbol{D}\boldsymbol{a} - \boldsymbol{e}_{\mathrm{RSEL}} = \boldsymbol{x} - \boldsymbol{D}\boldsymbol{a} - (\boldsymbol{e}_{\Omega} - \boldsymbol{u}) = \boldsymbol{e} - \boldsymbol{e}_{\Omega} + \boldsymbol{u} = \boldsymbol{e}_{\Lambda} + \boldsymbol{u}. \tag{14}$$

We define $\boldsymbol{v} = \boldsymbol{e}_{\Lambda} + \boldsymbol{u}$ where $\boldsymbol{e}_{\Lambda}$ and $\boldsymbol{u}$ are orthogonal. Moreover, $\boldsymbol{e}_{\Lambda}$ is from the driving noise, whereas $\boldsymbol{u}$ is introduced by the estimation error of the RSEL algorithm. Thus, $\boldsymbol{e}_{\Lambda}$ and $\boldsymbol{u}$ are uncorrelated. According to the Gauss–Markov theorem, we have the covariance for the solution $\boldsymbol{a}_{\mathrm{RSEL}}$ as

$$E[(\boldsymbol{a}_{\mathrm{RSEL}} - \boldsymbol{a}_{\mathrm{d}}))(\boldsymbol{a}_{\mathrm{RSEL}} - \boldsymbol{a}_{\mathrm{d}})^{\mathrm{T}}] = (\boldsymbol{D}^{\mathrm{T}}\boldsymbol{R}_v^{-1}\boldsymbol{D})^{-1}. \tag{15}$$

Similar to the method in [37], we firstly consider $N = P$ and then the matrix $\boldsymbol{D}$ is invertible. In this case, (13) can be written as

$$E[(\boldsymbol{a}_{\mathrm{LS}} - \boldsymbol{a}_{\mathrm{d}})(\boldsymbol{a}_{\mathrm{LS}} - \boldsymbol{a}_{\mathrm{d}})^{\mathrm{T}}] = \boldsymbol{D}^{-1}\boldsymbol{R}_e(\boldsymbol{D}^{-1})^{\mathrm{T}} \tag{16}$$

in which $\boldsymbol{R}_e = \boldsymbol{R}_{e_\Lambda} + \boldsymbol{R}_{e_\Omega}$. Similarly, regarding (15), we have

$$E[(\boldsymbol{a}_{\mathrm{RSEL}} - \boldsymbol{a}_{\mathrm{d}})(\boldsymbol{a}_{\mathrm{RSEL}} - \boldsymbol{a}_{\mathrm{d}})^{\mathrm{T}}] = \boldsymbol{D}^{-1}\boldsymbol{R}_v(\boldsymbol{D}^{-1})^{\mathrm{T}} \tag{17}$$

in which $\boldsymbol{R}_v = \boldsymbol{R}_u + \boldsymbol{R}_{e_\Lambda}$. In this case, we have

$$E[(\boldsymbol{a}_{\mathrm{LS}} - \boldsymbol{a}_{\mathrm{d}})(\boldsymbol{a}_{\mathrm{LS}} - \boldsymbol{a}_{\mathrm{d}})^{\mathrm{T}}] = \boldsymbol{D}^{-1}\boldsymbol{R}_{e_\Lambda}(\boldsymbol{D}^{-1})^{\mathrm{T}} + \boldsymbol{D}^{-1}\boldsymbol{R}_{e_\Omega}(\boldsymbol{D}^{-1})^{\mathrm{T}} \tag{18}$$

and

$$E[(\boldsymbol{a}_{\mathrm{RSEL}} - \boldsymbol{a}_{\mathrm{d}})(\boldsymbol{a}_{\mathrm{RSEL}} - \boldsymbol{a}_{\mathrm{d}})^{\mathrm{T}}] = \boldsymbol{D}^{-1}\boldsymbol{R}_{e_\Lambda}(\boldsymbol{D}^{-1})^{\mathrm{T}} + \boldsymbol{D}^{-1}\boldsymbol{R}_u(\boldsymbol{D}^{-1})^{\mathrm{T}}. \tag{19}$$

As $\boldsymbol{e}_\Omega = \boldsymbol{e}_{\mathrm{RSEL}}$,

$$\begin{aligned}&\mathrm{tr}(E[(\boldsymbol{a}_{\mathrm{LS}} - \boldsymbol{a}_{\mathrm{d}})(\boldsymbol{a}_{\mathrm{LS}} - \boldsymbol{a}_{\mathrm{d}})^{\mathrm{T}}] - E[(\boldsymbol{a}_{\mathrm{RSEL}} - \boldsymbol{a}_{\mathrm{d}})(\boldsymbol{a}_{\mathrm{RSEL}} - \boldsymbol{a}_{\mathrm{d}})^{\mathrm{T}}])\\ &\quad = \mathrm{tr}(\boldsymbol{D}^{-1}\boldsymbol{R}_{e_{\mathrm{RSEL}}}(\boldsymbol{D}^{-1})^{\mathrm{T}}).\end{aligned} \tag{20}$$

As $\boldsymbol{R}_{e_{\mathrm{RSEL}}}$ is a positive-definite matrix, $\boldsymbol{D}^{-1}\boldsymbol{R}_{e_{\mathrm{RSEL}}}(\boldsymbol{D}^{-1})^{\mathrm{T}}$ is also positive definite. Thus,

$$\mathrm{tr}(E[(\boldsymbol{a}_{\mathrm{LS}} - \boldsymbol{a}_{\mathrm{d}})(\boldsymbol{a}_{\mathrm{LS}} - \boldsymbol{a}_{\mathrm{d}})^{\mathrm{T}}]) > \mathrm{tr}(E[(\boldsymbol{a}_{\mathrm{RSEL}} - \boldsymbol{a}_{\mathrm{d}})(\boldsymbol{a}_{\mathrm{RSEL}} - \boldsymbol{a}_{\mathrm{d}})^{\mathrm{T}}]). \tag{21}$$

when $N > P$, a $P \times P$ invertible submatrix $\boldsymbol{D}_P$ of $\boldsymbol{D}$ exists based on the fact that $\boldsymbol{D}$ has a full-column rank. Therefore, we obtain

$$\boldsymbol{x}_P - \boldsymbol{D}_P\boldsymbol{a}_{\mathrm{LS}} = \boldsymbol{e}_P \tag{22}$$

and

$$\boldsymbol{x}_P - \boldsymbol{D}_P\boldsymbol{a}_{\mathrm{RSEL}} - \boldsymbol{e}_{\mathrm{RSEL},P} = \boldsymbol{v}_P \tag{23}$$

where $\boldsymbol{x}_P$, $\boldsymbol{e}_P$, $\boldsymbol{e}_{\mathrm{RSEL},P}$ and $\boldsymbol{v}_P$ represent corresponding subvectors of $\boldsymbol{x}$, $\boldsymbol{e}$, $\boldsymbol{e}_{\mathrm{RSEL}}$ and $\boldsymbol{v}$. In this case, following a similar approach as above, we find that

$$E[(\boldsymbol{a}_{\mathrm{LS}} - \boldsymbol{a}_{\mathrm{d}})(\boldsymbol{a}_{\mathrm{LS}} - \boldsymbol{a}_{\mathrm{d}})^{\mathrm{T}}] = \boldsymbol{D}_P^{-1}\boldsymbol{R}_{e_P}(\boldsymbol{D}_P^{-1})^{\mathrm{T}} \tag{24}$$

and

$$E[(\boldsymbol{a}_{\mathrm{RSEL}} - \boldsymbol{a}_{\mathrm{d}})(\boldsymbol{a}_{\mathrm{RSEL}} - \boldsymbol{a}_{\mathrm{d}})^{\mathrm{T}}] = (\boldsymbol{D}_P^{\mathrm{T}}\boldsymbol{R}_{\boldsymbol{v}_P}^{-1}\boldsymbol{D}_P)^{-1}. \tag{25}$$

Thus, in the same way, we have

$$\mathrm{tr}(E[(\boldsymbol{a}_{\mathrm{LS}} - \boldsymbol{a}_{\mathrm{d}})(\boldsymbol{a}_{\mathrm{LS}} - \boldsymbol{a}_{\mathrm{d}})^{\mathrm{T}}]) > \mathrm{tr}(E[(\boldsymbol{a}_{\mathrm{RSEL}} - \boldsymbol{a}_{\mathrm{d}})(\boldsymbol{a}_{\mathrm{RSEL}} - \boldsymbol{a}_{\mathrm{d}})^{\mathrm{T}}]).$$

Finally, we conclude that

$$E[\|\boldsymbol{a}_{\mathrm{RSEL}} - \boldsymbol{a}_{\mathrm{d}}\|_2^2] < E[\|\boldsymbol{a}_{\mathrm{LS}} - \boldsymbol{a}_{\mathrm{d}}\|_2^2]. \tag{26}$$

□

Robustness In the following, we present a theorem for the robustness of the solution $\boldsymbol{a}_{\text{RSEL}}$.

Theorem 2 *The solution $\boldsymbol{a}_{\text{RSEL}}$ of (10) is robust to a small disturbance for the sequential linear prediction model in (9).*

The method used in [37] can also be employed to prove Theorem 2. We omit the proof to save the space because it is similar to that in [37]. The difference lies in the constraint for the prediction error vector. Nonetheless, small disturbance cannot affect the sparsity of the prediction error vector, which guarantees the robustness of the solution $\boldsymbol{a}_{\text{RSEL}}$. In this paper, we assume that small disturbances arise from the mismatch between the training data at the training stage and the test data at the application stage. Thus, this theorem has a significant implication for the stability of our proposed system.

Convergence In Theorem 3, we show that the proposed algorithm can converge to a globally optimal solution provided a correctly identified support.

Theorem 3 *Suppose that the proposed algorithm can identify the support $\varOmega$ of the prediction error vector within finite iterations and* $\text{supp}(\boldsymbol{e}^{k+1}) = \text{supp}(\boldsymbol{e}^{k}) = \varOmega$. *If* $\beta \in (0, \frac{2}{3})$, *the RSEL algorithm converges to a globally optimal solution of* (10).

Proof Based on the ideal support identification of the proposed RSEL algorithm, the iterative equations can be written as

$$\boldsymbol{a}^{k+1} = \boldsymbol{a}^{k} + \beta(\boldsymbol{D}^{\mathrm{T}}\boldsymbol{e}^{k} + \boldsymbol{D}^{\mathrm{T}}\boldsymbol{x}_i - \boldsymbol{D}^{\mathrm{T}}\boldsymbol{D}\boldsymbol{a}^{k}) \tag{27}$$

$$\boldsymbol{e}^{k+1} = \boldsymbol{e}^{k} + \beta(P_{\varOmega}(\boldsymbol{D}\boldsymbol{a}^{k} - \boldsymbol{x}_i) - \boldsymbol{e}^{k}) \tag{28}$$

where the operator $P_{\varOmega}$ is defined as

$$P_{\varOmega}(\boldsymbol{u}) = \begin{cases} u_i, & \text{if } i \in \varOmega; \\ 0, & \text{Otherwise.} \end{cases} \tag{29}$$

The optimal solution of (10) can be denoted as $(\boldsymbol{a}^{*}, \boldsymbol{e}^{*})$ and $\text{supp}(\boldsymbol{e}^{*}) = \varOmega$. Regarding the orthogonality of the support, we have

$$(\boldsymbol{D}\boldsymbol{a}^{k} - \boldsymbol{x}_i - P_{\varOmega}(\boldsymbol{D}\boldsymbol{a}^{k} - \boldsymbol{x}_i))^{\mathrm{T}}(P_{\varOmega}(\boldsymbol{D}\boldsymbol{a}^{k} - \boldsymbol{x}_i) - \boldsymbol{e}^{*}) = 0 \tag{30}$$

$$(P_{\varOmega}(\boldsymbol{D}\boldsymbol{a}^{k} - \boldsymbol{x}_i) - \boldsymbol{e}^{*})^{\mathrm{T}}(\boldsymbol{e}^{*} - \boldsymbol{D}\boldsymbol{a}^{*} + \boldsymbol{x}_i) = 0 \tag{31}$$

Adding (30) and (31), we have

$$(P_{\varOmega}(\boldsymbol{D}\boldsymbol{a}^{k} - \boldsymbol{x}_i) - \boldsymbol{e}^{*})^{\mathrm{T}}(\boldsymbol{D}(\boldsymbol{a}^{k} - \boldsymbol{a}^{*}) + \boldsymbol{e}^{*} - P_{\varOmega}(\boldsymbol{D}\boldsymbol{a}^{k} - \boldsymbol{x}_i)) = 0. \tag{32}$$

And then we get

$$\begin{aligned}(P_{\varOmega}(\boldsymbol{D}\boldsymbol{a}^{k} - \boldsymbol{x}_i) - \boldsymbol{e}^{k} + \boldsymbol{e}^{k} - \boldsymbol{z}^{*})^{\mathrm{T}}(\boldsymbol{D}(\boldsymbol{a}^{k} - \boldsymbol{a}^{*})\\ + \boldsymbol{e}^{*} - \boldsymbol{e}^{k} + \boldsymbol{e}^{k} - P_{\varOmega}(\boldsymbol{D}\boldsymbol{a}^{k} - \boldsymbol{x}_i)) = 0.\end{aligned} \tag{33}$$

It follows that

$$\begin{aligned}
&-\left\|\boldsymbol{e}^k-\boldsymbol{e}^*\right\|_2^2-\left\|P_\Omega(\boldsymbol{D}\boldsymbol{a}^k-\boldsymbol{x}_i)-\boldsymbol{e}^k\right\|_2^2\\
&\quad=2(\boldsymbol{e}^k-\boldsymbol{e}^*)^{\mathrm{T}}(P_\Omega(\boldsymbol{D}\boldsymbol{a}^k-\boldsymbol{x}_i)-\boldsymbol{e}^k)-(\boldsymbol{e}^k-\boldsymbol{e}^*)^{\mathrm{T}}\boldsymbol{D}(\boldsymbol{a}^k-\boldsymbol{a}^*)\\
&\qquad-(P_\Omega(\boldsymbol{D}\boldsymbol{a}^k-\boldsymbol{x}_i)-\boldsymbol{e}^k)^{\mathrm{T}}\boldsymbol{D}(\boldsymbol{a}^k-\boldsymbol{a}^*).
\end{aligned}\tag{34}$$

As in [37], we introduce a symmetric and positive-definite matrix operator $\boldsymbol{H}$ as

$$\boldsymbol{H}=\begin{bmatrix}\boldsymbol{D}^{\mathrm{T}}\boldsymbol{D}+\boldsymbol{I}_{P\times P} & -\boldsymbol{D}^{\mathrm{T}}\\ -\boldsymbol{D} & 2\boldsymbol{I}_{P\times P}\end{bmatrix}.$$

Let us define a new vector $\boldsymbol{z}^k=\left[\boldsymbol{a}^{k\mathrm{T}}\ \boldsymbol{e}^{k\mathrm{T}}\right]^{\mathrm{T}}$, so that (27) and (28) can be transformed into

$$\begin{aligned}
\boldsymbol{z}^{k+1}&=\boldsymbol{z}^k+\beta\begin{bmatrix}\boldsymbol{D}^{\mathrm{T}}\boldsymbol{e}^k+\boldsymbol{D}^{\mathrm{T}}\boldsymbol{x}_i-\boldsymbol{D}^{\mathrm{T}}\boldsymbol{D}\boldsymbol{a}^k\\ (\boldsymbol{D}\boldsymbol{a}^k-\boldsymbol{x}_i)-\boldsymbol{e}^k\end{bmatrix}\\
&=\boldsymbol{z}^k+\beta F(\boldsymbol{z}^k).
\end{aligned}\tag{35}$$

Since $\boldsymbol{H}$ is symmetric and positive definite, there must exist another symmetric and positive-definite matrix $\boldsymbol{H}_1$ such that $\boldsymbol{H}_1^2=\boldsymbol{H}$. Hence, we obtain

$$\begin{aligned}
\left\|\boldsymbol{H}_1(\boldsymbol{z}^{k+1}-\boldsymbol{z}^*)\right\|_2^2&=\left\|\boldsymbol{H}_1(\boldsymbol{z}^k+\beta F(\boldsymbol{z}^k)-\boldsymbol{z}^*)\right\|_2^2\\
&=\left\|\boldsymbol{H}_1(\boldsymbol{z}^k+\beta F(\boldsymbol{z}^k)-\boldsymbol{z}^*)\right\|_2^2+\beta^2\left\|\boldsymbol{H}_1(F(\boldsymbol{z}^k)))\right\|_2^2\\
&\quad+2\beta(\boldsymbol{z}^k-\boldsymbol{z}^*)^{\mathrm{T}}\boldsymbol{H}F(\boldsymbol{z}^k).
\end{aligned}\tag{36}$$

where $\boldsymbol{z}^*=\left[\boldsymbol{a}^{*\mathrm{T}}\ \boldsymbol{e}^{*\mathrm{T}}\right]^{\mathrm{T}}$. Applying (34) to (36), we have

$$\begin{aligned}
\left\|\boldsymbol{H}_1(\boldsymbol{z}^{k+1}-\boldsymbol{z}^*)\right\|_2^2&=\left\|\boldsymbol{H}_1(\boldsymbol{z}^k-\boldsymbol{z}^*)\right\|_2^2+\beta^2\left\|\boldsymbol{H}_1F(\boldsymbol{z}^k)\right\|_2^2\\
&\quad-2\beta(\left\|F(\boldsymbol{z}^k)\right\|_2^2+\left\|\boldsymbol{D}(\boldsymbol{a}^k-\boldsymbol{a}^*)\right\|_2^2+\left\|\boldsymbol{e}^k-\boldsymbol{e}^*\right\|_2^2)\\
&\le\left\|\boldsymbol{H}_1(\boldsymbol{z}^k-\boldsymbol{z}^*)\right\|_2^2+\beta^2\left\|\boldsymbol{H}_1F(\boldsymbol{z}^k)\right\|_2^2-2\beta\left\|F(\boldsymbol{z}^k)\right\|_2^2.
\end{aligned}\tag{37}$$

Since $\|\boldsymbol{H}\|_2^2\le 3$, it follows that

$$\left\|\boldsymbol{H}_1(\boldsymbol{z}^{k+1}-\boldsymbol{z}^*)\right\|_2^2\le\left\|\boldsymbol{H}_1(\boldsymbol{z}^k-\boldsymbol{z}^*)\right\|_2^2+(3\beta^2-2\beta)\left\|F(\boldsymbol{z}^k)\right\|_2^2.\tag{38}$$

As $0<\beta\le\frac{2}{3}$, we get

$$\left\|\boldsymbol{H}_1(\boldsymbol{z}^{k+1}-\boldsymbol{z}^*)\right\|_2^2\le\left\|\boldsymbol{H}_1(\boldsymbol{z}^k-\boldsymbol{z}^*)\right\|_2^2.$$

In this case, the RSEL algorithm can converge to a globally optimal solution of the optimization problem in (10). □

3.3 Clustering Algorithm and Dictionary Construction

Upon extracting the corresponding prediction coefficient vectors of the training speech using the above-proposed RSEL algorithm, we need to employ a clustering algorithm to obtain the codebook of prediction coefficients for the voiced and unvoiced speech, respectively. In traditional speech processing methods, the Linde–Buzo–Gray (LBG) algorithm [1] is typically employed to provide codebooks for speech-relevant parameters. The codebook generation method may have a significant impact on the performance of the learned dictionary. However, it is very difficult to directly design the clustering algorithm based on the feedback of the reconstruction error in the application stage. Alternatively, minimizing the distance between the LPC vectors and the cluster centroids provides a simple and effective criterion for the clustering algorithm. To further improve the clustering performance, we therefore propose to use the K nearest neighbors (KNN) algorithm [3], as a substitute for the partition step in the LBG algorithm. The resulting combinational algorithm, called NNLBG in this paper, is expected to provide high-quality codebooks for the dictionary construction, as further explained below.

In view of the sequential linear prediction model in (9), we have

$$\boldsymbol{x}_i = \boldsymbol{\psi}(\boldsymbol{e}_i + \boldsymbol{G}\boldsymbol{x}_{i-1}) \tag{39}$$

where $\boldsymbol{G} = \begin{bmatrix} \boldsymbol{0}_{P\times(N-P)} & \boldsymbol{B} \\ \boldsymbol{0}_{(N-P)\times(N-P)} & \boldsymbol{0}_{(N-P)\times P} \end{bmatrix}$ with $\boldsymbol{B} = \begin{bmatrix} a_i(P) & \ldots & a_i(1) \\ & \ddots & \vdots \\ 0 & & a_i(P) \end{bmatrix}$, and

$$\boldsymbol{\psi} = \begin{bmatrix} 1 & 0 & 0 & \cdots & \cdots & & & 0 \\ -a_i(1) & 1 & 0 & \cdots & \cdots & & & 0 \\ -a_i(2) & -a_i(1) & 1 & \cdots & \cdots & & & 0 \\ \vdots & \vdots & \vdots & \ddots & & & & \vdots \\ -a_i(P) & -a_i(P-1) & \cdots & -a_i(1) & 1 & \cdots & & 0 \\ 0 & -a_i(P) & \cdots & \cdots & & \ddots & & 0 \\ \vdots & \vdots & \cdots & & & & & \\ 0 & 0 & \cdots & & -a_i(P) & \cdots & -a_i(1) & 1 \end{bmatrix}^{-1} \tag{40}$$

is a lower triangular matrix, constructed with the LPC vector. The matrix $\boldsymbol{G}$ is sparse, and $\boldsymbol{G}\boldsymbol{a}_i$ is a P-sparse vector. As the prediction error vector $\boldsymbol{e}_i$ is also sparse, the ith training speech signal is sparse with respect to the basis $\boldsymbol{\psi}$. The clustering algorithm NNLBG is applied to the prediction coefficient vectors of the training speech signals, which are estimated through the proposed RSEL algorithm in Sect. 3.2, to obtain the corresponding codebooks $\boldsymbol{Q}^{\mathrm{v}} = \left[\boldsymbol{q}_1^{\mathrm{v}}\ \boldsymbol{q}_2^{\mathrm{v}} \cdots \boldsymbol{q}_c^{\mathrm{v}}\right]$ and $\boldsymbol{Q}^{\mathrm{u}} = \left[\boldsymbol{q}_1^{\mathrm{u}}\ \boldsymbol{q}_2^{\mathrm{u}} \cdots \boldsymbol{q}_c^{\mathrm{u}}\right]$, respectively, for voiced and unvoiced speech, where c is the number of clusters.

It is well known that the dictionary can be constructed from a union of bases [15]. Specifically, we can fill the column vector $\boldsymbol{q}_j^{\mathrm{v}}(j = 1, 2, \ldots, c)$ and $\boldsymbol{q}_j^{\mathrm{u}}(j = 1, 2, \ldots, c)$ within the above structured matrix $\boldsymbol{\psi}$ in (40) to generate multiple bases $\boldsymbol{\psi}_j^{\mathrm{v}}(j = 1, 2, \ldots, c)$ and $\boldsymbol{\psi}_j^{\mathrm{u}}(j = 1, 2, \ldots, c)$ for both cases of voiced and unvoiced speech, respectively. The dictionaries $\boldsymbol{\Psi}^{\mathrm{v}}$ and $\boldsymbol{\Psi}^{\mathrm{u}}$ can then be expressed as follows,

$$\boldsymbol{\Psi}^{\mathrm{v}} = \left[\boldsymbol{\psi}_1^{\mathrm{v}} \quad \boldsymbol{\psi}_2^{\mathrm{v}} \quad \cdots \quad \boldsymbol{\psi}_c^{\mathrm{v}}\right] \tag{41}$$

$$\boldsymbol{\Psi}^{\mathrm{u}} = \left[\boldsymbol{\psi}_1^{\mathrm{u}} \quad \boldsymbol{\psi}_2^{\mathrm{u}} \quad \cdots \quad \boldsymbol{\psi}_c^{\mathrm{u}}\right] \tag{42}$$

These two dictionaries are structured, easy to implement and can reduce the storage cost because only the codebook including Pc entries, instead of the dictionary composed of Nc entries, needs to be stored. In contrast with the reference dictionary learning methods, our proposed dictionaries are learned with LPC features instead of the raw speech signals. They hold the advantages of both analytic and data-driven dictionaries [23]: It is easy and efficient to deploy; it can yield effective sparse representation; and they are stable and robust to noise in compressive speech sensing applications.

3.4 Recovery Algorithm

In the application stage, the recovery algorithm module is fed with the received measurements, the sensing matrix and the dictionary for effective sparse reconstruction. In this section, we discuss two cases: the noise-free case and the noise-aware case.

The noise-free case refers to an ideal environment where the received observations are the same as the transmitted ones. In this case, assuming that the input speech signal in the ith frame is voiced, the received measurement vector can be expressed as

$$\boldsymbol{y}_i = \boldsymbol{\Phi}\boldsymbol{x}_i = \boldsymbol{\Phi}\boldsymbol{\Psi}^{\mathrm{v}}(\boldsymbol{e}_i + \boldsymbol{s}_i) \tag{43}$$

where $\boldsymbol{s}_i$ is a Pc-sparse vector and can be estimated as

$$\boldsymbol{s}_i = \frac{\lambda_i}{c}\boldsymbol{G}^{\mathrm{v}}\hat{\boldsymbol{x}}_{i-1} \tag{44}$$

where λ_i is a regularization factor to reduce error propagation, and matrix $\boldsymbol{G}^{\mathrm{v}}$ is the expansion of the matrix $\boldsymbol{G}$ in (39) and can be generated by filling the codebook $\boldsymbol{Q}^{\mathrm{v}}$ to $\boldsymbol{G}$, i.e., $\boldsymbol{G}^{\mathrm{v}} = \left[\boldsymbol{G}_{\boldsymbol{q}_1^{\mathrm{v}}}^{\mathrm{T}}\ \boldsymbol{G}_{\boldsymbol{q}_2^{\mathrm{v}}}^{\mathrm{T}} \cdots \boldsymbol{G}_{\boldsymbol{q}_c^{\mathrm{v}}}^{\mathrm{T}}\right]^{\mathrm{T}}$, and $\hat{\boldsymbol{x}}_{i-1}$ is the recovered speech signal from the previous frame with the index $i-1$. Therefore, the following optimization technique can be employed to estimate the sparse coefficient vector of the ith speech frame,

$$\hat{\boldsymbol{e}}_i = \arg\min \|\boldsymbol{e}_i\|_0 \qquad \text{s.t.} \quad \boldsymbol{y}_i - \boldsymbol{\Phi}\boldsymbol{\Psi}^{\mathrm{v}}\boldsymbol{s}_i = \boldsymbol{\Phi}\boldsymbol{\Psi}^{\mathrm{v}}\boldsymbol{e}_i. \tag{45}$$

With the estimated sparse coefficient vector $\hat{\boldsymbol{e}}_i$ from (45), we can reconstruct the ith speech frame as

$$\hat{\boldsymbol{x}}_i = \boldsymbol{\Psi}^{\mathrm{v}}(\hat{\boldsymbol{e}}_i + \boldsymbol{s}_i).$$

Similarly, for the unvoiced speech, we can fill the matrix $\boldsymbol{G}$ with the codebook $\boldsymbol{Q}^{\mathrm{u}}$ to construct the matrix $\boldsymbol{G}^{\mathrm{u}}$ as $\boldsymbol{G}^{\mathrm{u}} = \left[\boldsymbol{G}^{\mathrm{T}}_{\boldsymbol{q}^{\mathrm{u}}_1}\ \boldsymbol{G}^{\mathrm{T}}_{\boldsymbol{q}^{\mathrm{u}}_2}\ \cdots\ \boldsymbol{G}^{\mathrm{T}}_{\boldsymbol{q}^{\mathrm{u}}_c}\right]^{\mathrm{T}}$. The sparse coefficient vector of the jth unvoiced speech frame can be estimated through

$$\boldsymbol{s}_j = \frac{\lambda_j}{c}\boldsymbol{G}^{\mathrm{u}}\hat{\boldsymbol{x}}_{j-1} \tag{46}$$

$$\hat{\boldsymbol{e}}_j = \arg\min \left\|\boldsymbol{e}_j\right\|_0 \quad \text{s.t.} \quad \boldsymbol{y}_j - \boldsymbol{\Phi}\boldsymbol{\Psi}^{\mathrm{u}}\boldsymbol{s}_j = \boldsymbol{\Phi}\boldsymbol{\Psi}^{\mathrm{u}}\boldsymbol{e}_j. \tag{47}$$

Then, the unvoiced speech for the jth speech frame can be reconstructed as

$$\hat{\boldsymbol{x}}_j = \boldsymbol{\Psi}^{\mathrm{u}}(\hat{\boldsymbol{e}}_j + \boldsymbol{s}_j).$$

The optimization problems in (45) and (46) can be solved through the BP algorithm or the OMP algorithm. Since we utilize the information from the previous speech frame in sparse reconstruction, the method proposed above is referred to as a sequential recovery algorithm. In a real environment, noise might be inevitable to some extent. The noise-aware case refers to that measurements are corrupted by the noise, which is here modeled as an additive white Gaussian disturbance. In this case, the above optimization problems in (45) and (46) should be, respectively, modified as

$$\min \|\boldsymbol{e}_i\|_0 \qquad \text{s.t.} \quad \left\|\boldsymbol{y}_i - \boldsymbol{\Phi}\boldsymbol{\Psi}^{\mathrm{v}}\boldsymbol{s}_i - \boldsymbol{\Phi}\boldsymbol{\Psi}^{\mathrm{v}}\boldsymbol{e}_i\right\|_2 \leq \epsilon \tag{48}$$

$$\min \left\|\boldsymbol{e}_j\right\|_0 \qquad \text{s.t.} \quad \left\|\boldsymbol{y}_j - \boldsymbol{\Phi}\boldsymbol{\Psi}^{\mathrm{v}}\boldsymbol{s}_j - \boldsymbol{\Phi}\boldsymbol{\Psi}^{\mathrm{v}}\boldsymbol{e}_j\right\|_2 \leq \epsilon \tag{49}$$

where the threshold $\epsilon > 0$ is used to control the noise energy. These two optimization problems can be solved through the basis pursuit denoising (BPDN) algorithm [9] or the OMP algorithm [32].

4 Experimental Evaluation

In this section, the performance of our proposed system is evaluated with the speech dataset from the GRID corpus [26], which is freely available to researchers. This corpus consists of recordings of 1000 sentences for each of 34 speakers (18 male speakers and 16 female speakers). All the utterances were normalized to have a maximum absolute magnitude of 1. We randomly select 10 speakers of both genders for our experiments. All the speech signals in both training dataset and validation dataset are downsampled to 16KHz. We use segmental signal-to-noise ratio (SSNR) [18], perceptual evaluation of speech quality (PESQ) [18] score and short-time objective intelligibility (STOI) [29] score as the objective measures to evaluate the reconstructed speech quality of our proposed system. Moreover, the performance of our proposed system is evaluated in comparison with the KSVD algorithm [2], MOD algorithm [12], ODL algorithm [21] and PCA algorithm [25]. For the experiments below, the parameters involved are set as follows: prediction order $P = 4$, sparsity level $S = 20$, stopping criterion $\eta = 10^{-6}$, number of clusters $c = 3$. All the experiments were conducted in MATLAB R2018a (64 bit) on a desktop computer with an Intel i7-8700 CPU (3.2 GHz) and 16 GHz RAM.

4.1 Speaker-Dependent Case in Noise-Free Environment

In the speaker-dependent case, the dictionaries for the voiced and unvoiced frames are, respectively, learned with the training dataset from each speaker at the application stage. Subsequently, the evaluation of the system performance is performed separately for each speaker at the application stage. Three speakers of both genders are involved in the experiments of this part. For each speaker, 20 utterances are selected randomly for the training stage and another 10 utterances, guaranteed to be distinct from the training set, are utilized at the application stage. It should be noted that the training dataset for the KSVD, MOD, ODL and PCA methods includes 70 utterances from each speaker in order to guarantee sufficient speech data are exploited to train atoms of these dictionaries. The final results regarding the SSNR and the PESQ scores of the reconstructed speech signals are averaged over all the speakers involved. Moreover, in noise-free environment, the evaluation is implemented with both the BP and OMP algorithms, the most representative recovery algorithms in CS, to reconstruct speech signals at the application stage.

In Table 2, we compare the average SSNR results for male speakers among the above-mentioned five methods, namely: KSVD, MOD, ODL, PCA and the proposed method. It is conspicuous that our proposed system can achieve better performance than the other four methods under all the compression rates and for both recovery algorithms. For instance, with the BP algorithm, with a compression rate $M/N = 0.5$, the average SSNR of our proposed system amounts to 20.9 dB, with the improvement to the other four methods ranging from 5.1 dB to 8.9 dB. Similarly, at the same compression rate, with the OMP algorithm, our proposed system can achieve an average SSNR at 16.2 dB; the corresponding improvements with the other four methods range

Table 2 Average SSNR (dB) for male speakers in speaker-dependent case when using dictionaries from KSVD, MOD, ODL, PCA and the proposed technique

M/N	BP					OMP				
	KSVD	MOD	ODL	PCA	Proposed	KSVD	MOD	ODL	PCA	Proposed
0.3	11.7	11.4	11.5	7.20	14.1	6.61	7.06	6.54	3.97	8.10
0.4	13.7	13.0	13.4	9.63	17.5	7.93	8.17	7.62	6.34	11.6
0.5	15.8	14.7	15.6	12.0	20.9	10.9	9.50	10.4	8.52	16.2
0.6	18.2	16.5	18.1	14.6	24.2	13.5	11.4	12.9	10.3	18.9
0.7	20.9	18.6	20.7	17.5	27.3	15.2	13.1	14.8	11.6	20.5
0.8	24.2	21.2	23.7	21.1	30.8	16.5	14.6	16.2	12.6	21.5

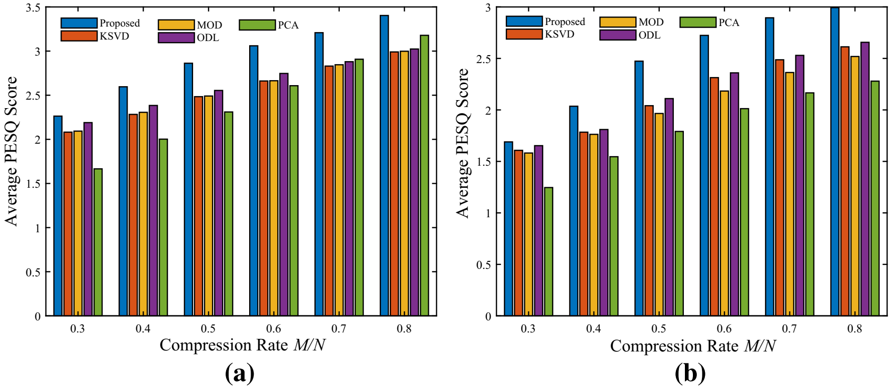


Fig. 2 Average PESQ scores for male speakers in speaker-dependent case when using dictionaries from KSVD, MOD, ODL, PCA and the proposed technique. Recovered by **a** BP algorithm, **b** OMP algorithm

Table 3 Average SSNR (dB) for female speakers in speaker-dependent case when using dictionaries from KSVD, MOD, ODL, PCA and the proposed technique

M/N	BP					OMP				
	KSVD	MOD	ODL	PCA	Proposed	KSVD	MOD	ODL	PCA	Proposed
0.3	10.8	11.3	10.9	7.40	13.0	6.32	7.04	6.22	4.00	6.94
0.4	12.3	12.7	12.4	9.54	16.3	7.13	8.01	7.07	6.12	9.90
0.5	13.9	14.1	14.4	11.7	19.7	9.25	9.35	9.77	8.14	14.4
0.6	16.0	15.7	16.8	14.2	22.9	11.6	11.2	12.4	9.81	16.9
0.7	20.9	18.6	20.7	16.9	27.3	13.4	13.0	14.2	11.1	18.6
0.8	21.7	20.0	22.8	20.2	29.2	14.8	14.4	15.5	12.2	19.7

from 5.4 dB to 7.7 dB. The average PESQ scores for the male speakers in this case are illustrated in Fig. 2. The histograms clearly show that our proposed system can achieve better PESQ scores. For example, at the compression rate of 0.5, our proposed method can improve the average PESQ score of the reconstructed speech with the BP algorithm from 2.31 (PCA) to 2.86, and when the reconstruction is done with the OMP algorithm, the PESQ is enhanced from 1.79 (PCA) to 2.47.

The experimental results of average SSNR and PESQ scores for female speakers are demonstrated in Table 3 and Fig. 3. It is clear that our proposed technique can also achieve better performance for female speakers. The improvement of our proposed method on the average SSNR at compression rate of 0.5 with BP algorithm is over 5 dB, while the average PESQ score is increased from 2.28 to 2.80 with the proposed method. We can also find that the OMP algorithm can achieve average SSNR and PESQ gains with our proposed method of around 5 dB and no less than 0.3, respectively.

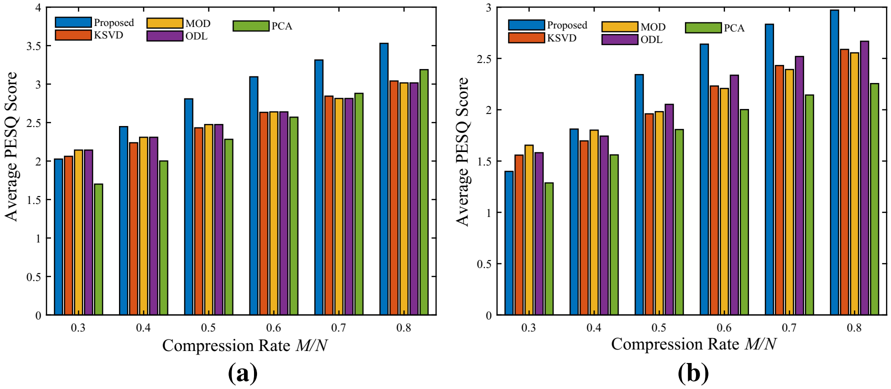


Fig. 3 Average PESQ scores for female speakers in speaker-dependent case when using dictionaries from KSVD, MOD, ODL, PCA and the proposed technique. Recovered by **a** BP algorithm, **b** OMP algorithm

4.2 Speaker-Dependent Case in Noise-Aware Environment

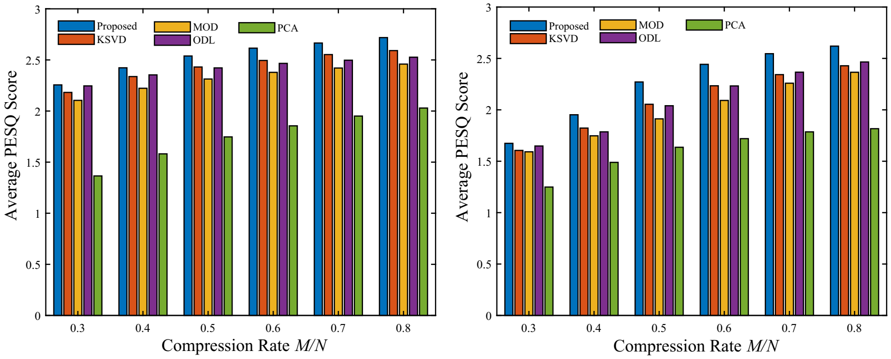
This subsection studies the system performance in the speaker-dependent case when the received measurement vectors are corrupted by the additive Gaussian white noise with zero mean and standard deviation 0.002. Apart from the SSNR and PESQ score, we add the STOI score to evaluate the intelligibility of the recovered speech signals in the presence of noise. Although it is well acknowledged that the background noise can deteriorate the recovery performance of CS, our proposed system can still achieve better performance than the other four approaches. Tables 4 and 5 show the average SSNR results for male and female speakers, respectively, and the maximum improvement of the proposed method over the other four methods at the compression rate of 0.5 for both genders is around 8 dB. Figure 4a, b, respectively, gives the average PESQ scores of the reconstructed male speech from the BPDN and the OMP in noise-aware environment based on different dictionaries trained with various approaches. These results clearly show the improvement of our proposed technique over the benchmark approaches. For example, at the compression rate of 0.5, the average PESQ score is

Table 4 Average SSNR (dB) for male speakers in the presence of noise when using dictionaries from KSVD, MOD, ODL, PCA and the proposed technique

M/N	BPDN					OMP				
	KSVD	MOD	ODL	PCA	Proposed	KSVD	MOD	ODL	PCA	Proposed
0.3	9.62	8.02	8.04	3.72	11.4	4.85	5.45	4.92	4.03	5.59
0.4	10.5	8.74	8.63	4.64	12.8	6.10	5.93	5.19	5.70	7.25
0.5	11.1	9.20	9.01	5.30	13.8	8.27	6.64	7.21	6.82	10.5
0.6	11.5	9.55	9.26	5.84	14.4	9.96	7.98	9.27	7.44	12.4
0.7	11.7	9.78	9.44	6.28	15.0	11.1	9.38	10.79	7.91	13.7
0.8	12.0	9.97	9.59	6.60	15.3	12.0	10.5	12.0	8.23	14.6

Table 5 Average SSNR (dB) for female speakers in the presence of noise when using dictionaries from KSVD, MOD, ODL, PCA and the proposed technique

M/N	BP					OMP				
	KSVD	MOD	ODL	PCA	Proposed	KSVD	MOD	ODL	PCA	Proposed
0.3	9.56	8.54	8.51	4.12	10.9	5.32	6.04	5.23	4.44	5.35
0.4	10.4	9.20	9.06	4.98	12.3	6.11	6.54	5.54	6.00	7.03
0.5	11.0	9.64	9.42	5.64	13.3	7.96	7.30	7.65	7.00	10.4
0.6	11.4	9.93	9.64	6.14	14.0	9.60	8.62	9.86	7.66	12.50
0.7	11.7	9.19	9.85	6.55	14.5	10.8	10.0	11.4	8.07	13.8
0.8	11.9	10.4	9.98	6.90	14.9	11.8	11.2	12.5	8.42	14.7

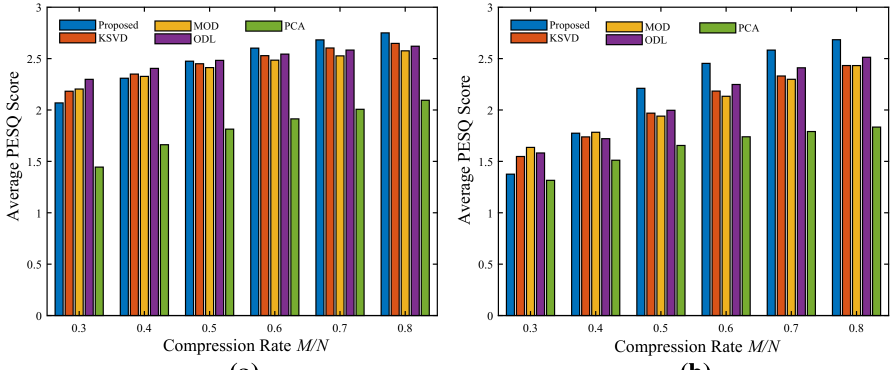


(a) **(b)**

Fig. 4 Average PESQ scores for male speakers in speaker-dependent case in the presence of noise when using dictionaries from KSVD, MOD, ODL, PCA and the proposed technique. Recovered by **a** BPDN algorithm, **b** OMP algorithm

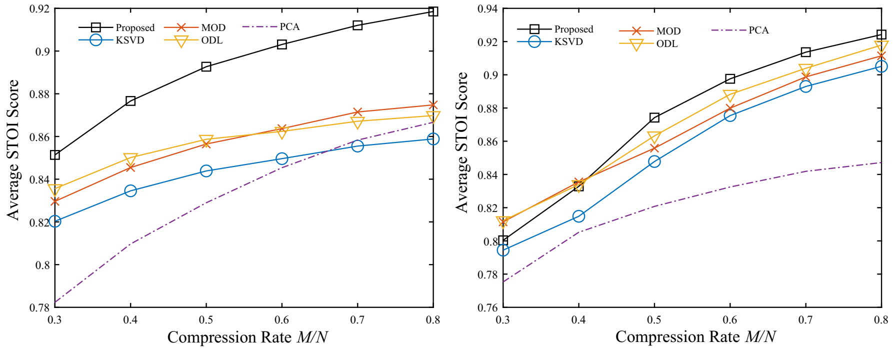
improved through the proposed method from 1.75 (PCA) to 2.54 in the case of the BPDN algorithm, while for the OMP algorithm, it is improved from 1.64 (PCA) to 2.27. The PESQ score of the reconstructed female speech from the BPDN at the compression rate of 0.5 in Fig. 5a is improved from 1.81 (PCA) to 2.47. Figure 5b shows that our proposed method can improve the average PESQ score of the reconstructed female speech with the OMP from 1.65 (PCA) to 2.21.

The intelligibility of the recovered speech signals is evaluated through the average STOI scores in Figs. 6 and 7, respectively, for male and female speakers. When $M = 0.5N$ and the BPDN is utilized as the recovery algorithm, the average STOI scores with the KSVD are 0.84 and 0.86, respectively, for male and female speakers. The ODL achieves a slightly better performance than the KSVD with the average STOI scores for both genders increased to 0.86. The MOD achieves nearly the same average STOI score for male speakers as the ODL, while the one for female speakers is improved to 0.87. The average STOI scores of the PCA for male and female speakers are 0.83 and 0.82, respectively. The average STOI scores for both genders with our proposed method reach higher values of 0.89 and 0.90, respectively. With the OMP as the



(a) **(b)**

Fig. 5 Average PESQ scores for female speakers in speaker-dependent case in the presence of noise when using dictionaries from KSVD, MOD, ODL, PCA and the proposed technique. Recovered by **a** BPDN algorithm, **b** OMP algorithm



(a) **(b)**

Fig. 6 Average STOI scores for male speakers in speaker-dependent case in noisy environment when using dictionaries from KSVD, MOD, ODL, PCA and the proposed method. Recovered by **a** BPDN algorithm, **b** OMP algorithm

recovery algorithm, our proposed technique can improve the average STOI scores from 0.82 to 0.87 and 0.88, respectively, for male and female speakers. Based on these experimental results, we conclude that our proposed system is more robust to background noise than the state-of-the-art methods under comparison.

4.3 Speaker-Independent Case in Noise-Free Environment

In the speaker-independent case, we utilize speech signals of the above 6 speakers as the training data, while 10 utterances of another two speakers of both genders are randomly selected from the GRID corpus as the test dataset at the application stage, i.e., ten speakers are involved in the experiments of this subsection. The average SSNR results in the speaker-independent case are presented in Tables 6 and 7, respectively, for male and female speakers. It can be clearly observed that our proposed method

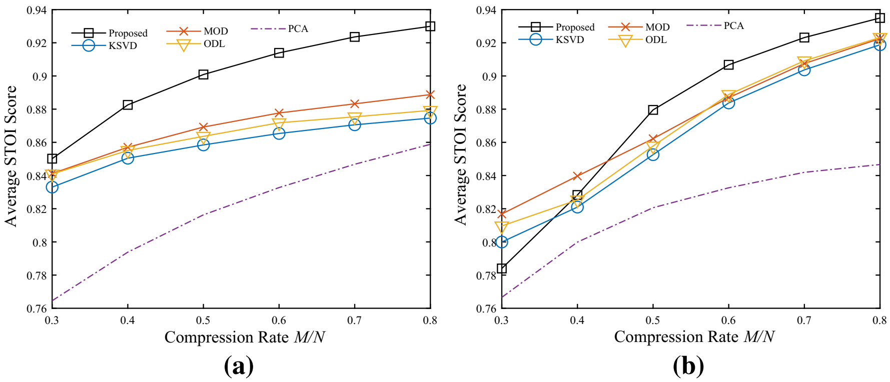


Fig. 7 Average STOI scores for female speakers in speaker-dependent case in noisy environment when using dictionaries from KSVD, MOD, ODL, PCA and the proposed method. Recovered by **a** BPDN algorithm, **b** OMP algorithm

Table 6 Average SSNR (dB) for male speakers in speaker-independent case when using dictionaries from KSVD, MOD, ODL, PCA and the proposed technique

M/N	BP					OMP				
	KSVD	MOD	ODL	PCA	Proposed	KSVD	MOD	ODL	PCA	Proposed
0.3	15.4	14.2	14.0	7.07	17.1	9.34	8.63	8.23	3.89	10.1
0.4	17.5	15.8	15.5	10.3	20.2	11.5	9.75	9.04	7.69	13.3
0.5	19.9	17.3	16.8	13.3	23.5	15.5	11.1	10.6	10.9	18.4
0.6	22.5	18.9	18.2	16.2	26.5	18.2	13.0	12.8	13.2	21.2
0.7	25.4	20.6	19.7	19.2	29.5	19.7	14.7	14.6	15.1	22.7
0.8	28.5	22.9	22.0	22.7	32.7	20.8	16.3	16.1	16.4	23.8

Table 7 Average SSNR (dB) for female speakers in speaker-independent case when using dictionaries from KSVD, MOD, ODL, PCA and the proposed technique

M/N	BP					OMP				
	KSVD	MOD	ODL	PCA	Proposed	KSVD	MOD	ODL	PCA	Proposed
0.3	12.2	10.8	10.8	7.38	14.2	7.00	5.97	5.79	4.41	7.82
0.4	14.4	12.4	12.3	10.0	17.6	9.15	7.08	6.63	7.42	10.7
0.5	17.0	14.0	13.7	12.6	21.0	13.3	8.48	8.17	10.0	15.8
0.6	19.6	15.6	15.1	15.2	24.3	15.9	10.4	10.4	12.2	18.5
0.7	22.4	17.5	16.8	18.0	27.4	17.4	12.2	12.3	13.9	20.0
0.8	25.6	19.9	19.4	21.4	30.7	18.5	13.7	13.8	15.2	21.0

achieves better average SSNR than the reference approaches regardless of the recovery algorithms and the genders. For example, with the BP algorithm, the average SSNRs of the reconstructed speech of male speakers at the compression rate of 0.6 for ODL, MOD, KSVD and PCA are 18.2 dB, 18.9 dB, 22.5 dB and 16.2 dB, respectively. Our

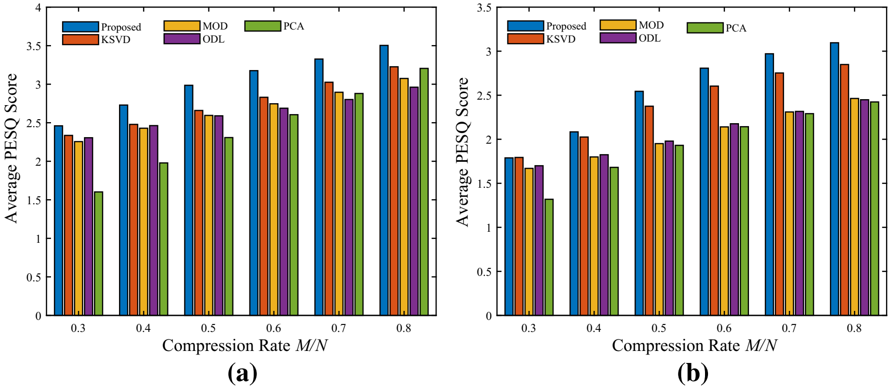


Fig. 8 Average PESQ scores for male speakers in speaker-independent case when using dictionaries from KSVD, MOD, ODL, PCA and the proposed technique. Recovered by **a** BP algorithm, **b** OMP algorithm

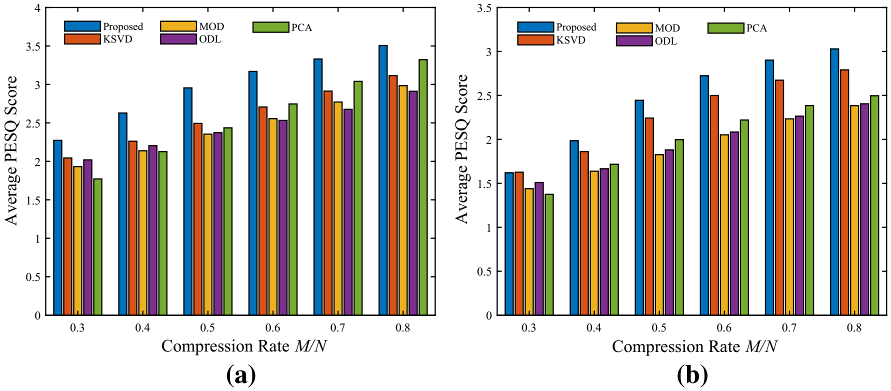


Fig. 9 Average PESQ scores for female speakers in speaker-independent case when using dictionaries from KSVD, MOD, ODL, PCA and the proposed technique. Recovered by **a** BP algorithm, **b** OMP algorithm

proposed method can improve the performance to 26.5 dB. Meanwhile, the improvement of our proposed approach with respect to average PESQ scores, as shown in Figs. 8 and 9 for both genders, is also conspicuous. For instance, as illustrated in Fig. 8b, our proposed technique can improve the average PESQ scores from 2.14 to 2.80 at the compression rate of 0.6. Therefore, our proposed method can obtain higher-quality speech signals than the other four methods in the speaker-independent case.

4.4 Speaker-Independent Case in Noise-Aware Environment

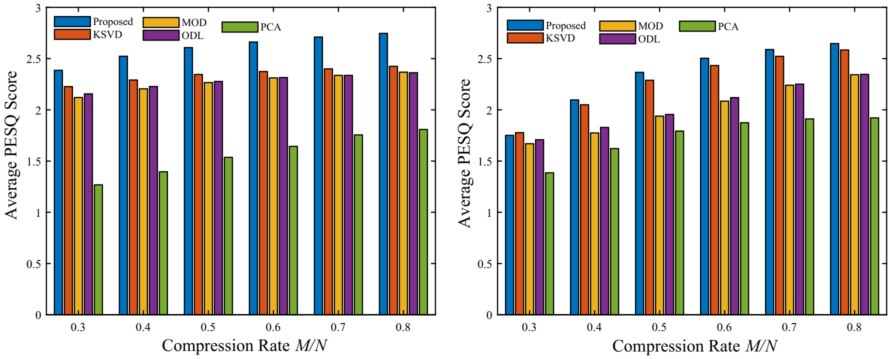
We consider the speaker-independent case when the measurement vectors at the application stage are corrupted by additive white Gaussian noise. The average SSNR results for male and female speakers under this scenario are presented in Tables 8 and 9. For instance, at the compression rate of 0.6, KSVD with BPDN as the recovery algorithm can achieve the largest average SSNR of 11.9 dB for male speakers among the exist-

Table 8 Average SSNR (dB) for male speakers in speaker-independent case in the presence of noise when using dictionaries from KSVD, MOD, ODL, PCA and the proposed technique

M/N	BPDN					OMP				
	KSVD	MOD	ODL	PCA	Proposed	KSVD	MOD	ODL	PCA	Proposed
0.3	10.5	9.63	9.49	3.17	14.8	7.83	7.57	7.30	4.45	8.41
0.4	11.1	10.4	10.2	4.31	16.1	10.0	8.19	7.62	7.11	11.3
0.5	11.6	11.0	10.7	5.43	17.0	12.7	9.00	8.58	8.66	14.1
0.6	11.9	11.3	11.0	6.31	17.6	14.4	10.4	10.3	9.45	15.7
0.7	12.0	11.6	11.2	7.11	18.1	15.5	11.8	11.9	9.95	16.7
0.8	12.2	11.8	11.4	7.75	18.4	16.2	12.9	13.0	10.3	17.3

Table 9 Average SSNR (dB) for female speakers in speaker-independent case in the presence of noise when using dictionaries from KSVD, MOD, ODL, PCA and the proposed technique

M/N	BPDN					OMP				
	KSVD	MOD	ODL	PCA	Proposed	KSVD	MOD	ODL	PCA	Proposed
0.3	8.43	7.50	7.45	3.37	12.0	5.29	4.91	4.73	4.74	5.99
0.4	9.09	8.31	8.17	4.46	13.5	7.74	5.53	5.01	6.97	9.11
0.5	9.47	8.81	8.59	5.43	14.5	10.4	6.38	6.12	8.21	11.9
0.6	9.72	9.17	8.90	6.23	15.2	12.2	7.82	7.85	8.95	13.5
0.7	9.94	9.43	9.11	7.00	15.8	13.3	9.29	9.38	9.42	14.5
0.8	10.0	9.66	9.32	7.63	16.1	14.0	10.5	10.7	9.78	15.2



(a) **(b)**

Fig. 10 Average PESQ scores for male speakers in speaker-independent case in the presence of noise when using dictionaries from KSVD, MOD, ODL, PCA and the proposed technique. Recovered by **a** BPDN algorithm, b OMP algorithm

ing four reference methods. However, the average SSNR of our proposed approach reaches 17.6 dB. The 5.7-dB increment indicates that it can achieve better performance in noise reduction.

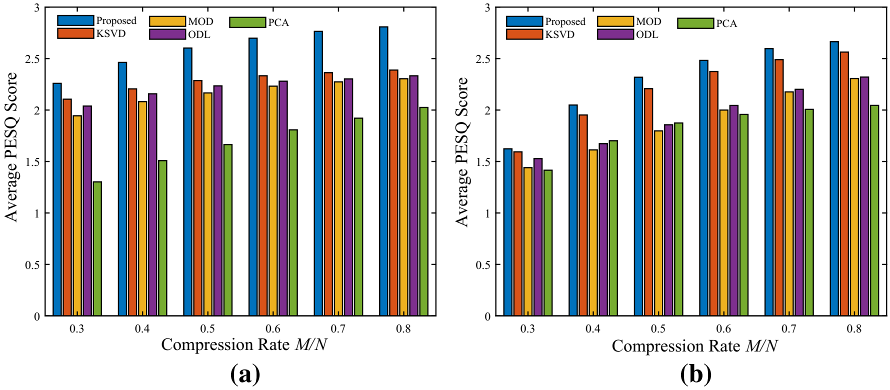


Fig. 11 Average PESQ scores for female speakers in speaker-independent case in the presence of noise when using dictionaries from KSVD, MOD, ODL, PCA and the proposed technique. Recovered by **a** BPDN algorithm, **b** OMP algorithm

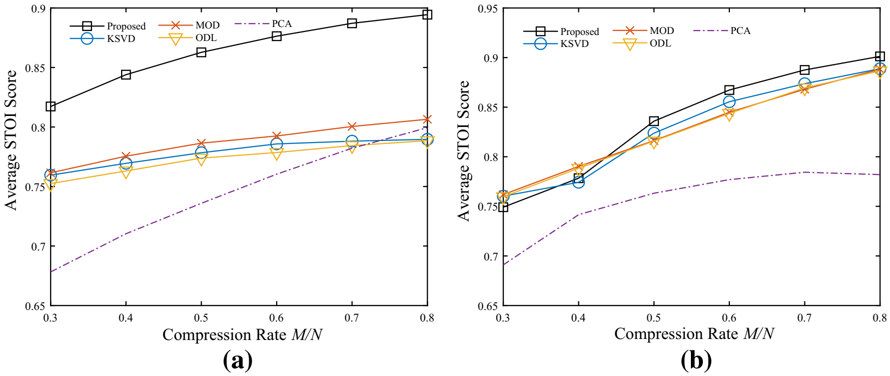


Fig. 12 Average STOI scores for male speakers in speaker-independent case in the presence of noise when using dictionaries from KSVD, MOD, ODL, PCA and the proposed technique. Recovered by **a** BPDN algorithm, **b** OMP algorithm

The average PESQ results in the speaker-independent case in the presence of noise are presented in Figs. 10 and 11, respectively, for both genders. The average PESQ scores of our proposed method are higher than the other methods. As observed in Fig. 11b, at the compression rate of 0.6, our proposed technique can improve the average PESQ score from 1.87 to 2.48. Meanwhile, the average STOI scores are given in Figs. 12 and 13, respectively, for male and female speakers. The clear improvement in average STOI scores with our method indicates that it can produce more intelligible speech signals. As illustrated in Fig. 13a, at the compression rate of 0.6, our proposed approach improves the average STOI score from 0.82 to 0.90 and can thus reduce the background noise more effectively than the other four methods in a speaker-independent case.

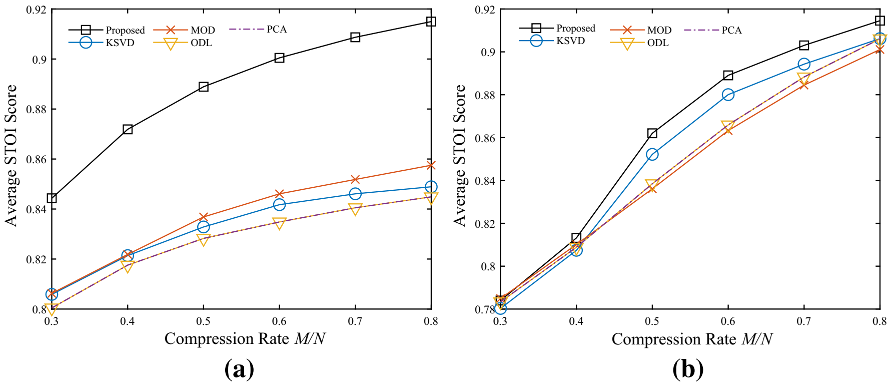


Fig. 13 Average STOI scores for female speakers in speaker-independent case in the presence of noise when using dictionaries from KSVD, MOD, ODL, PCA and the proposed technique. Recovered by **a** BPDN algorithm, **b** OMP algorithm

5 Conclusion

In this paper, we have presented a new compressive speech sensing system which is composed of two stages, namely the training stage and the application stage. The core of training stage is the RNN-based dictionary learning module which learns structured dictionaries for both voiced and unvoiced speech. In particular, we leveraged the sequential linear prediction model and the proposed RSEL to extract the speech LPCs and applied the NNLBG algorithm to cluster the LPC vectors in order to generate effective codebooks. Then, the dictionaries for voiced and unvoiced speech were constructed with a union of bases obtained from the column vectors in corresponding codebooks. Moreover, we provided a theoretical analysis of the mean squared error, robustness and convergence of the proposed RSEL algorithm. In the application stage, a sequential recovery algorithm was proposed to reconstruct speech signals. It was shown through an extensive experimental study that our proposed system can outperform the state-of-the-art methods in both speaker-dependent and speaker-independent cases under the noise-free as well as noise-aware conditions.

Acknowledgements This work is supported in part by the Natural Sciences and Engineering Research Council of Canada, the National Natural Science Foundation of China (Grant Nos. 61601248, 61771263, 61871241) and the University Natural Science Research Foundation of Jiangsu Province, China (Grant No. 16KJB510037).

References

1. C.C. Aggarwal, C.K. Reddy, *Data Clustering: Algorithms and Applications* (CRC Press, New York, 2013), pp. 60–65
2. M. Aharon, M. Elad, A. Bruckstein, K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation. IEEE Trans. Signal Process. **54**(11), 4311–4322 (2006)
3. N.S. Altman, An introduction to kernel and nearest-neighbor nonparametric regression. Am. Stat. **46**(3), 175–185 (1992)

4. C.L. Bao, H. Ji, Y.H. Quan, Z.W. Shen, Dictionary learning for sparse coding: algorithms and convergence analysis. IEEE Trans. Pattern Anal. Mach. Intell. **38**(7), 1356–1369 (2016)
5. E.J. Candes, M.B. Wakin, An introduction to compressive sampling. IEEE Signal Process. Mag. **25**(2), 20–21 (2008)
6. E.J. Candes, J.K. Romberg, T. Tao, Stable signal recovery from incomplete and inaccurate measurements. Commun. Pure Appl. Math. **59**(8), 1207–1223 (2006)
7. E.J. Candes, T. Tao, Near-optimal signal recovery from random projections: universal encoding strategies? IEEE Trans. Inf. Theory **52**(12), 5406–5425 (2006)
8. E.J. Candes, J. Romberg, T. Tao, Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. IEEE Trans. Inf. Theory **52**(2), 489–509 (2006)
9. S.S. Chen, D.L. Donoho, M.A. Saunders, Atomic decomposition by basis pursuit. SIAM Rev. **43**(1), 129–159 (2001)
10. D.L. Donoho, Compressed sensing. IEEE Trans. Inf. Theory **52**(4), 1289–1306 (2006)
11. Y.C. Eldar, G. Kutyniok, *Compressed Sensing: Theory and Applications* (Cambridge University Press, New York, 2012), pp. 20–25
12. K. Engan, S.O. Aase, J.H. Husoy, Multi-frame compression: theory and design. Signal Process. **80**(10), 2121–2140 (2000)
13. S. Foucart, H. Rauhut, *A Mathematical Introduction to Compressive Sensing* (Birkhauser, New York, 2013), pp. 40–50
14. D. Giacobello, M.G. Christensen, M.N. Murthi, S.H. Jensen, M. Moonen, Sparse linear prediction and its applications to speech processing. IEEE Trans. Audio Speech Lang. Process. **20**(5), 1610–1644 (2012)
15. R. Gribonval, M. Nielsen, Sparse representations in unions of bases. IEEE Trans. Inf. Theory **49**(12), 3320–3325 (2003)
16. A. Hosseini, J. Wang, S.M. Hosseini, A recurrent neural network for solving a class of generalized convex optimization problems. Neural Netw. **44**, 78–86 (2013)
17. X.L. Hu, J. Wang, A recurrent neural network for solving a class of general variational inequalities. IEEE Trans. Syst. Man Cybern. B (Cybern.) **37**(3), 528–539 (2007)
18. Y. Hu, P.C. Loizou, Evaluation of objective quality measures for speech enhancement. IEEE Trans. Audio Speech Lang. Process. **16**(1), 229–238 (2008)
19. J.N. Laska, P.T. Boufounos, M.A. Davenport, R.G. Baraniuk, Democracy in action: quantization, saturation, and compressive sensing. Appl. Comput. Harmon. Anal. **31**(3), 429–443 (2011)
20. S.H. Liu, Y.D. Zhang, T. Shan, R. Tao, Structure-aware Bayesian compressive sensing for frequency-hopping spectrum estimation with missing observations. IEEE Trans. Signal Process. **66**(8), 2153–2166 (2018)
21. J. Mairal, F. Bach, J. Ponce, G. Sapiro, Online learning for matrix factorization and sparse coding. J. Mach. Learn. Res. **11**(1), 19–60 (2010)
22. D. Needle, J.A. Tropp, CoSaMP: Iterative signal recovery from incomplete and inaccurate samples. Appl. Comput. Harmon. Anal. **26**(3), 301–321 (2009)
23. R. Rubinstein, M. Zibulevsky, M. Elad, Double sparsity: learning sparse dictionaries for sparse signal approximation. IEEE Trans. Signal Process. **58**(3), 1553–1564 (2010)
24. S.J. Sengijpta, *Fundamentals of Statistical Signal Processing: Estimation Theory* (Taylor and Francis Group, Abingdon, 1995), pp. 100–105
25. P. Sharma, V. Abrol, A.D. Dileep, A.K. Sao, Sparse coding based features for speech units classification. Comput. Speech Lang. **47**, 333–350 (2018)
26. C.D. Sigg, T. Dikk, J.M. Buhmann, Speech enhancement using generative dictionary learning. IEEE Trans. Audio Speech Lang. Process. **20**(6), 1698–1712 (2012)
27. P. Stoica, R.L. Moses, *Spectral Analysis of Signals* (Pearson Prentice Hall, Upper Saddle River, 2005), pp. 80–90
28. L.H. Sun, Z. Yang, Y.Y. Ji, L. Ye, Reconstruction of compressed speech sensing based on overcomplete linear prediction dictionary. Chin. J. Sci. Instrum. **4**, 733–739 (2012)
29. C.H. Taal, R.C. Hendriks, R. Heusdens, J. Jensen, An algorithm for intelligibility prediction of time-frequency weighted noisy speech. IEEE Trans. Audio Speech Lang. Process. **19**(7), 2125–2136 (2011)
30. D. Tank, J. Hopfield, Simple neural optimization networks: An A/D converter, signal decision circuit, and a linear programming circuit. IEEE Trans. Circuits Syst. **33**(5), 533–541 (1986)
31. I. Tosic, P. Frossard, Dictionary learning. IEEE Signal Process. Mag. **28**(2), 27–38 (2011)

32. J.A. Tropp, A.C. Gilbert, Signal recovery from random measurements via orthogonal matching pursuit. IEEE Trans. Inf. Theory **53**(12), 4655–4666 (2007)
33. T.H. Vu, V. Monga, Fast low-rank shared dictionary learning for image classification. IEEE Trans. Image Process. **26**(11), 5160–5175 (2017)
34. J.C. Wang, Y.S. Lee, C.H. Lin, S.F. Wang, C.H. Shih, C.H. Wu, Compressive sensing-based speech enhancement. IEEE/ACM Trans. Audio Speech Lang. Process. **24**(11), 2122–2131 (2016)
35. D.L. Wu, W.P. Zhu, M. Swamy, The theory of compressive sensing matching pursuit considering time-domain noise with application to speech enhancement. IEEE/ACM Trans. Audio Speech Lang. Process. **22**(3), 682–696 (2014)
36. Y.S. Xia, M.S. Kamel, A generalized least absolute deviation method for parameter estimation of autoregressive signals. IEEE Trans. Neural Netw. **19**(1), 107–118 (2008)
37. Y.S. Xia, M.S. Kamel, H. Leung, A fast algorithm for AR parameter estimation using a novel noise-constrained least-squares method. Neural Netw. **23**(3), 396–405 (2010)
38. Y.S. Xia, J. Wang, Low-dimensional recurrent neural network-based Kalman filter for speech enhancement. Neural Netw. **67**, 131–139 (2015)
39. Z. Zhang, Y. Xu, J. Yang, X.L. Li, D. Zhang, A survey of sparse representation: algorithms and applications. IEEE Access **3**, 490–500 (2015)