Paper

Incoherent Discriminative Dictionary Learning for Speech Enhancement

Dima Shaheen, Oumayma Al Dakkak, and Mohiedin Wainakh

Telecommunications Department, Higher Institute for Applied Science and Technology HIAST, Damascus, Syria

https://doi.org/10.26636/jtit.2018.121317

Abstract—Speech enhancement is one of the many challenging tasks in signal processing, especially in the case of non-stationary speech-like noise. In this paper a new incoherent discriminative dictionary learning algorithm is proposed to model both speech and noise, where the cost function accounts for both "source confusion" and "source distortion" errors, with a regularization term that penalizes the coherence between speech and noise sub-dictionaries. At the enhancement stage, we use sparse coding on the learnt dictionary to find an estimate for both clean speech and noise amplitude spectrum. In the final phase, the Wiener filter is used to refine the clean speech estimate. Experiments on the Noizeus dataset, using two objective speech enhancement measures: frequency-weighted segmental SNR and Perceptual Evaluation of Speech Quality (PESQ) demonstrate that the proposed algorithm outperforms other speech enhancement methods tested.

Keywords—ADMM, l1 minimization algorithms, sparse coding, speech enhancement, supervised dictionary learning.

1. Introduction

Digital speech is a communication tool that is most frequently used by humans, especially with the proliferation of Voice over the Internet (VoIP) telephony software. Speech can be corrupted by various factors: noise (additive, multiplicative), reverberation (convolutive noise), and interfering speech. Speech enhancement aims to boost its quality. This enhancement involves two quality factors: "speech pleasantness", which refers to how comfortable it is for humans to listen to the speech signal over a prolonged period of time, and "speech intelligibility", which refers to how understandable the speech is (word error rate). Noise is the most common factor that causes speech degradation. Speech de-noising algorithms constitute a major part of the enhancement methods that aim to extract a clean speech signal from a noisy mix. It is a challenging task, as it is hard to remove noise efficiently without distorting the clean signal.

The problem we are tackling in this paper is single channel speech de-noising that deals with non-stationary noise. Mathematically, this problem aims to reconstruct the clean speech signal $s(n)$, based on the received signal $y(n)$ which is an additive mixture of the two unknown signals: the clean speech and a non-stationary noise signal $i(n)$:

$$y(n) = s(n) + i(n)\,. \tag{1}$$

The significance of this problem is based on the fact that communication takes place, nowadays, in noisy environments, such as at airports, in the street or inside a car. The noise in these environments is non-stationary, which means that its statistic values are changing over time. It is crucial to provide the user with a good quality speech, so they can understand others and listen to them comfortably, using communication tools, in these hostile environments. In fact, there are many applications that use speech de-noising algorithms in these adverse environments, such as mobile communications, VoIP, hearing aids and speech recognition software.

Traditional speech enhancement methods, like spectral subtraction (SS) [1], [2], Wiener filtering [3], statistical model-based methods [4] and subspace methods Singular Spectrum Analysis (SSA) [5], [6] perform well in the case of white noise, but have limited performance in the case of non-stationary speech-like noise. SS is based on estimating the noise power spectrum and subtracting it from the noisy power spectrum. The main issue with SS is the generation of isolated peaks in the estimated clean speech spectrum, which is referred to as musical noise. Statistical model-based methods assume that speech and noise obey some probability distribution and propose a least square estimator to estimate the signal. In both cases it is hard to find a good estimate for the noise power spectrum in the case of non-stationary noise. All these methods are unsupervised, which means that they do not use any prior information about the noise and speech. Recently, new supervised methods incorporating prior information to build a model for both speech and noise signals using training samples, have been proposed. These methods achieve better results than non-supervised methods.

Codebook-based approaches [7], [8], Hidden Markov Model (HMM) based approaches [9]–[11], supervised non-negative matrix factorization (NMF) [12]–[14] and sparsity-based method [15]–[19] are examples of supervised speech enhancement approaches. Srinivasan *et al.* [8] used vector quantization to learn codebooks for both speech and noise

LPC features. At the enhancement stage, the closest pair in terms of minimum Itakaru-Saito distance between the noisy power spectrum and linear combination of the speech and noise pair is picked and used as estimators. Mohammadabad *et al*. [14] proposed Bayesian NMF for speech enhancement, where the training data is decomposed into two matrices: bases matrix and activation matrix, while at the enhancement stage the noisy mixture is projected on the concatenation of the two matrices.

Motivated by the great success of the sparsity based signal model achieved in many signal processing tasks, and notably image de-noising [20], Sigg [15] proposed using the approximate K-Singular Value Decomposition (K-SVD) [21], [22] dictionary learning to model the amplitude spectrum of clean speech and noise separately, and then concatenating both dictionaries in one to perform speech enhancement.

Zhao *et al*. [16] proposed using the same K-SVD with a non-negative constraint at the sparse coding stage to learn a dictionary that models the Power Spectral Density (PSD) of clean speech, and used the Least Angle Regression LARS algorithm [34] to find the sparse code of the noisy speech on the learned dictionary. Then, the clean speech PSD estimate is found based on multiplication of the sparse code with the dictionary. Luo *et al*. [17] proposed a complementary joint sparse representation, where two mixture dictionaries to model "mixture and speech" and "mixture and noise" are added to the Generative Dictionary Learning (GDL) problem formulation, and sparse codes of clean speech are forced to represent the noisy mixture on the mixture and clean speech sub-dictionary, while the sparse codes for the noise are forced to represent the noisy mixture on the mixture and noise sub-dictionary. Though this joint sparse representation alleviates, to some extent, the problem of source confusion, it is characterized by high complexity due to the need of learning four sub-dictionaries instead of two.

In the previous studies, "signal approximation" only is considered in the cost function when learning the representative dictionaries, while *source confusion* and *incoherence* between speech and noisy sub-dictionaries are not taken into account in the dictionary learning process. Source confusion means that part of the noise that is coherent with clean speech will have sparse representation over the clean speech dictionary (noise confusion), and part of the clean speech will have sparse representation over the noise dictionary (speech confusion), and thus, residual noise corresponding to noise confusion might still exist in the estimated clean speech at the enhancement stage, which will also suffer from extra distortion from the original clean speech due to the fact that part of it will be omitted as it will be considered as noise. Incoherence refers to the maximum correlation between any two columns of speech and noise dictionaries. As shown in [15], incoherence is directly related to the degree of sparsity (number of non-zero elements) needed for the speech and noise signals to achieve exact recovery by their sparse projection on their corresponding sub-dictionaries. High coherence means a low sparsity degree, which cannot be verified in practice, and thus we are interested in low coherence dictionary.

In this paper, we propose a new Incoherent Discriminative Dictionary Learning (IDDL) algorithm to model both speech and noise jointly. We impose a coherence penalty on the speech and noise sub-dictionaries in the cost function, which also incorporates: a penalty for "speech confusion" when learning the noise sub-dictionary, and a penalty for "noise confusion" when learning the clean speech sub- dictionary. We use the Alternating Direction Method of Multipliers (ADMM) [35] to solve the two sub-dictionaries' learning optimization problems.

The paper is organized as follows. In Section 2 a review of the main problems is provided: dictionary learning algorithms and speech enhancement using sparse coding. In Section 3, the proposed IDDL algorithm is described, along with the overall proposed speech enhancement system. In Section 4, the conducted experiments and their results are presented. Section 5 summarizes and concludes the paper.

2. Problem Review

$\mathbf{Y} \in \mathbb{R}^{N\times n}$ is the matrix whose columns are the n training samples $\mathbf{y}_i \in \mathbb{R}^N$ (N is the dimension of the input signals. In the context of speech enhancement, the input signals $\mathbf{y}_i$ are the amplitude spectrum of every speech frame i, and thus, N is the number of FFT coefficients[1]), $\mathbf{D} \in \mathbb{R}^{N\times K}$ is the dictionary matrix whose columns are K prototype signals that can represent signals of interests *sparsely* (i.e. using a linear combination of a low number of these prototype signals denoted by $\mathbf{d}_i$), $\mathbf{X} \in \mathbb{R}^{k\times n}$ is the matrix whose columns $x_i \in \mathbb{R}^K$ are the sparse codes of $\mathbf{y}_i$. In our setting, $\mathbf{Y}$ contains the extracted features (amplitude of FFT coefficients) of the training audio frames (either clean speech or noise), composed of $\mathbf{Y}_s \in \mathbb{R}^{N\times n_s}$ speech training samples (n_s is the number of clean speech training frames), and $\mathbf{Y}_n \in \mathbb{R}^{N\times n_n}$ the noise training samples (n_n is the number of noise training frames). $\mathbf{X}_s \in \mathbb{R}^{K\times n_s}$ is the sparse codes of $\mathbf{Y}_s$, and $\mathbf{X}_n \in \mathbb{R}^{K\times n_n}$ the sparse codes of $\mathbf{Y}_n$. $\mathbf{D}$ is the concatenation of $\mathbf{D}_s \in \mathbb{R}^{N\times L}$ and $\mathbf{D}_n \in \mathbb{R}^{N\times L}$ the dictionary matrices for representation of the clean speech signal and the noise signal, respectively. They have the same number of columns denoted by L. Clearly in this case $= 2L$, and the total number of training samples $n = n_s + n_n$. $\mathbf{X}_s^s$ are the sparse coefficients of $\mathbf{X}_s$ on $\mathbf{D}_s$, $\mathbf{X}_s^n$ are the sparse coefficients of $\mathbf{X}_s$ on $\mathbf{D}_n$.$\mathbf{X}_n^n$ are the sparse coefficients of $\mathbf{X}_n$ on $\mathbf{D}_n$, and $\mathbf{X}_n^s$ are the sparse coefficients of $\mathbf{X}_n$ on $\mathbf{D}_s$.

2.1. Dictionary Learning

Sparsity-based signal model approximates a signal by a linear combination of a few basic signals out of a larger collection of signals that form what is called the dictionary. In

[1] In fact we take only half of the number of FFT coefficients because of symmetry, so $N = \frac{NFFT}{2} + 1$.

a classic dictionary learning problem, we seek a matrix $\mathbf{D}$ whose columns are the basic signals that can represent, as close as possible, the training signals $\mathbf{y}_i$ *sparsely*:

$$\min_{\mathbf{D},\mathbf{X}} \sum_{i=1}^{n} \left\|\mathbf{y}_i - \mathbf{D}\mathbf{x}_i\right\|_2^2 \quad \text{s.t.} \quad \forall i, \left\|\mathbf{x}_i\right\|_0 \leq k \ , \tag{2}$$

where k is the maximum number of non-zero elements in $\mathbf{x}_i$, $\|\mathbf{x}_i\|_0$ is ℓ_0 a pseudo norm which represents the number of non-zeros in $\mathbf{x}_i$. $\mathbf{X}$ is the matrix composed of all the sparse codes $\mathbf{x}_i$.

This optimization problem is non-convex when both $\mathbf{D}$ and $\mathbf{X}$ are unknown, however it becomes convex if one of $\mathbf{D}$ or $\mathbf{X}$ is fixed – that is why it is generally solved iteratively by fixing the dictionary $\mathbf{D}$ and updating sparse codes $\mathbf{X}$, and then fixing $\mathbf{X}$ and updating $\mathbf{D}$.

In fact, dictionary learning is a generalization of the k-means clustering algorithm [21], the only difference is that in k-means, each training signal is forced to use only one "atom" from the dictionary (the closest cluster center), as its representative, while in dictionary learning each signal is allowed to use multiple dictionary atoms, provided that it can be approximated by a linear combination of these atoms, and that this linear combination uses as few the dictionary atoms as possible.

In k-means, we iterate between finding the representative of each training signal (the cluster center which is equivalent to the dictionary atom that minimizes the suitable metric distance), and updating the cluster centers. However, dictionary learning is solved by iterating between two stages [21]. First, the dictionary is fixed and the sparse code $\mathbf{x}_i$ for each training signal is calculated using any sparse coding solver. Then, the sparse code is fixed and the dictionary atoms are updated to minimize the cost function.

The method used to update the dictionary atoms is the key difference between individual dictionary learning algorithms. Some dictionary learning methods update, in each iteration, the whole set of atoms. This is the case in one of the early and simple dictionary learning solutions – Method of Optimal Direction (MOD) [23], which updates the whole dictionary using the closed form of the Mean Squared Error (MSE) estimator:

$$\mathbf{D} = \mathbf{Y}\mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1} \ . \tag{3}$$

Other dictionary learning algorithms update the dictionary atoms successively, one by one, as is the case in the very famous and successful dictionary learning algorithm known as K-Singular Value Decomposition (K-SVD) [21]. At the sparse coding stage, K-SVD uses greedy Orthogonal Matching Pursuit (OMP) [32], [33] to find the sparse code for each training sample. At the dictionary update stage, in turn, for each dictionary atom $\mathbf{d}_k$, K-SVD selects only those training samples that use this atom, which will be denoted as $\mathbf{x}^k$, and splits the representation error E into two components: the sparse representation on $\mathbf{d}_k$, and the residual error $\mathbf{E}_k$ that accounts for the sparse presentation error using all the dictionary atoms other than $\mathbf{d}_k$:

$$\begin{aligned}\mathbf{E} &= \left\|\mathbf{Y} - \mathbf{D}\mathbf{X}\right\|_F^2 = \left\|\mathbf{Y} - \sum_{i=1}^{K} \mathbf{d}_i \mathbf{x}_T^i\right\|_F^2 \\ &= \left\|\left(\mathbf{Y} - \sum_{\substack{i\neq k \\ i=1}} \mathbf{d}_i \mathbf{x}_T^i\right) - \mathbf{d}_k \mathbf{x}_T^k\right\|_F^2 = \left\|\mathbf{E}_k - \mathbf{d}_k \mathbf{x}_T^k\right\|_F^2 \ , \end{aligned} \tag{4}$$

where $\mathbf{x}_T^i$ represents the sparse coefficients corresponding to the atom $\mathbf{d}_i$, which is the i-th row of the matrix $\mathbf{X}$. As the rows $\mathbf{x}_T^k$ are all zeros except for the indexes of the test examples in $\mathbf{Y}$ that use atoms $\mathbf{d}_k$ $\mathbf{d}_k\mathbf{x}_T^k$ does not affect the whole $\mathbf{E}_k$, but only the *restricted* $\mathbf{E}_k^R$ which is composed of the columns of $\mathbf{E}_k$ that correspond to the examples that use $\mathbf{d}_k$.

To update $\mathbf{d}_k$ and $\mathbf{x}_T^k$ in a way that minimizes the restricted error $\mathbf{E}_k^R$ (which is the only part of the total error representation $\mathbf{E}$ that is affected by atom $\mathbf{d}_k$), K-SVD evaluates the Singular Value Decomposition (SVD) for $\mathbf{E}_k^R = \mathbf{U}\mathbf{\Delta}\mathbf{V}^T$ (where $\mathbf{U}$ and $\mathbf{V}$ are orthonormal matrices, and Δ is a diagonal matrix with non-negative elements on the diagonal known as *eigen* values), and updates $\mathbf{d}_k$ with the first column of $\mathbf{U}$, simultaneously updating the corresponding sparse coefficients $\mathbf{x}_T^k$ as the first column of $\mathbf{V}$ multiplied by $\mathbf{\Delta}(1,1)$ [21].

The cost function in Eq. (4) measures the representation power of dictionary $\mathbf{D}$ only. In the case of a classification task, discriminative power of the sparse code $\mathbf{x}$ should be considered. This leads to a new trend in dictionary learning algorithms called "discriminative" or "supervised" dictionary learning in which the cost function reflects both the representation and classification error. Suo [24] has proposed the most general formulation of the discriminative dictionary learning problem, given below:

$$\min_{\mathbf{D},\mathbf{X}} \sum_{i=1}^{n} \left(\left\|\mathbf{y}_i - \mathbf{D}\mathbf{x}_i\right\|_2^2 + \lambda_1 \left\|\mathbf{x}_i\right\|_1\right) + \lambda_2 f_{\mathbf{x}}(\mathbf{X}) + \lambda_3 f_{\mathbf{D}}(\mathbf{D}) \ , \tag{5}$$

where $f_{\mathbf{X}}(\mathbf{X})$ is a function that measures the discriminative power of the sparse codes $\mathbf{X}$, and $f_{\mathbf{D}}(\mathbf{D})$ is a function that measures the discrimination power of the atoms of $\mathbf{D}$.

Discriminative dictionary learning algorithms fall into one of three categories, depending on the values of λ_2, λ_3. In the first category ($\lambda_3 = 0$), a dictionary shared by all classes is learned, while forcing the sparse codes to be discriminative. For example, Mairal *et al*. [25] proposed to add a logistic loss function to the sparse code, as a discriminative measure. Zhang *et al*. proposed Discriminative K-SVD (D-KSVD) [26] that adds a linear regression term to learn a linear classifier on the sparse coefficients to the objective function in the dictionary learning problem formulation (6), while in the case of label consistent-KSVD (LC-KSVD) [27], a label consistency term is added that measures how consistent the sparse codes are with the class labels.

In the second category ($\lambda_2 = 0$), only the discriminative power of the dictionary atoms is considered. For example,

Ramirez *et al*. [28] proposed learning class-specific sub-dictionaries for each class with a structural incoherence penalty term to make the sub-dictionaries as independent as possible.

A hybrid discriminative dictionary learning forms the third category, where both the dictionary atoms and the sparse codes are forced to be discriminative ($\lambda_2 \neq 0,\ \lambda_3 \neq 0$). It is the case in the COPAR dictionary learning algorithm [29] and Fisher discriminative dictionary learning (FDDL) [30].

FDDL uses label information both in the dictionary update and sparse coding stage. In FDDL, the sparse codes of the training samples are forced to have low within-class scatter, but large between-class scatter. Also, each class-specific sub-dictionary is forced to have good reconstruction capability for the training samples from that class, but poor reconstruction capability for other classes. Therefore, both the representation residual and the representation coefficients of the query sample are discriminative. Thus, the dictionary learning optimization problem is formulated, in FDDL, as follows:

$$\min_{\mathbf{D},\mathbf{X}} \mathrm{r}(\mathbf{Y},\mathbf{D},\mathbf{X}) + \lambda_1 \left\|\mathbf{X}\right\|_1 + \lambda_2 f_{\mathbf{X}}(\mathbf{X}) \quad \text{s.t.} \quad \left\|\mathbf{d}_i\right\|_2 = 1, \quad \forall i \in \{1 \ldots L\} \ , \tag{6}$$

where $\mathrm{r}(\mathbf{Y},\mathbf{D},\mathbf{X})$ is a cost function that measures the discriminative power of dictionary $\mathbf{D}$, $\|\mathbf{X}\|_1$ is the sparsity inducing term, and $f_{\mathbf{X}}(\mathbf{X})$ is the cost function that measures the discriminative power of the sparse codes $\mathbf{X}$.

The cost function that imposes discrimination of atoms of dictionary $\mathbf{D}$ is defined as:

$$\mathrm{r}(\mathbf{Y}_i,\mathbf{D},\mathbf{X}_i) = \left\|\mathbf{Y}_i - \mathbf{D}\mathbf{X}_i\right\|_F + \left\|\mathbf{Y}_i - \mathbf{D}_i\mathbf{X}_i^i\right\|_F + \sum_{\substack{j=1 \\ j\neq i}}^{C} \left\|\mathbf{D}_j\mathbf{X}_i^j\right\|_F^2 , \tag{7}$$

where $\|\mathbf{A}\|_F = \sqrt{\sum_i \sum_j a_{ij}^2}$ is the Frobenius norm, C is the number of classes, $\mathbf{Y}_i$ is the matrix composed of a training sample of class i, $\mathbf{X}_i$ is their corresponding sparse codes over the total dictionary $\mathbf{D}$. $\mathbf{D}_j$ is the sub-dictionary representing the samples of class j. The first term in "r" represents the total representation error of samples $\mathbf{Y}_i$ (of class i) over the total dictionary $\mathbf{D}$, and the second term represents the representation error of $\mathbf{Y}_i$ over the i-class specific sub-dictionary $\mathbf{D}_i$, while the third term represents the contribution of sub-dictionaries other than $\mathbf{D}_i$ in the sparse representation of samples $\mathbf{Y}_i$, which should be small, as those samples belong to a different class, and it accounts for the confusion error in the case of source separation.

Function $f_{\mathbf{X}}(\mathbf{X})$ is a cost function that imposes discrimination on the sparse codes $\mathbf{X}$ according to the Fisher discrimination criterion, which means that the sparse codes $\mathbf{X}$ should have minimum within-class scatter denoted by $S_W(\mathbf{X})$, and maximum between-class scatter denoted by $S_B(\mathbf{X})$. A regularization term that shrinks $\|\mathbf{X}\|_F^2$ is added to make $f_{\mathbf{X}}(\mathbf{X})$ more smooth and convex [30]:

$$f_{\mathbf{X}}(\mathbf{X}) = \mathrm{tr}\big(S_W(\mathbf{X})\big) - \mathrm{tr}\big(S_B(\mathbf{X})\big) + \eta \left\|\mathbf{X}\right\|_F^2 , \tag{8}$$

where:

$$S_W(\mathbf{X}) = \sum_{i=1}^{C} \sum_{x_k \in \mathbf{X}_i} (\boldsymbol{x}_k - \boldsymbol{m}_i)(\boldsymbol{x}_k - \boldsymbol{m}_i)^T \ , \tag{9}$$

$$S_B(\mathbf{X}) = \sum_{i=1}^{C} n_i (\boldsymbol{m}_i - \boldsymbol{m})(\boldsymbol{m}_i - \boldsymbol{m})^T \ . \tag{10}$$

In the above equations, $\boldsymbol{m}_i$ is the mean of sparse codes $\mathbf{X}_i$, $\boldsymbol{m}$ is the mean of all sparse vectors $\mathbf{X}$, n_i is the number of samples that belong to class i, η is a regularization parameter that controls the energy of the samples, tr is the matrix trace operator.

2.2. Speech Enhancement using Sparse Coding

Sigg [15] proposed a supervised speech enhancement method based on learning two dictionaries, one for clean speech and the other for noise, according to the following formulations:

$$\min_{\mathbf{D}_s,\mathbf{X}_s} \left\|\mathbf{Y}_s - \mathbf{D}_s\mathbf{X}_s\right\|_F^2 \ , \quad \|\mathbf{X}_s\|_0 \leq k_s \ , \tag{11}$$

$$\min_{\mathbf{D}_n,\mathbf{X}_n} \left\|\mathbf{Y}_n - \mathbf{D}_n\mathbf{X}_n\right\|_F^2 \ , \quad \|\mathbf{X}_n\|_0 \leq k_n \, . \tag{12}$$

Sigg proposed GDL to solve each of the previous problems. GDL is, in fact, a variation of the approximate K-SVD [22], the only difference is at the sparse coding stage. Sigg proposed least angle regression with coherence criterion (LARC) [15] for sparse coding, instead of the greedy orthogonal matching pursuit (OMP) [33]. LARC is a variation of LARS [34], where the coherence between the residual error and the dictionary is used as the stopping criterion instead of the l_2 norm of the residual error. It has been found that LARC has several advantages over OMP. First, LARC is insensitive to changes in signal energy, as the stopping criterion is related to residual coherence and not to the amplitude of the error. Second, as LARC uses the l_1 norm, which not only penalizes the num-

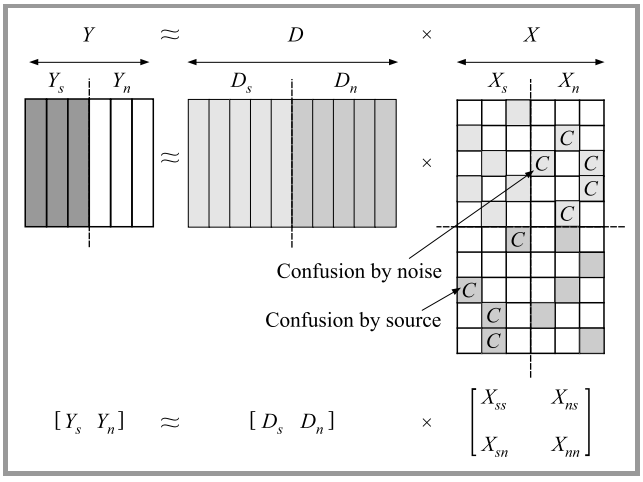

Fig. 1. Columns of $\mathbf{X}$ are the sparse codes of $\mathbf{Y}$ on dictionary $\mathbf{D}$. The sparse coefficient where there is C, means coefficients which cause confusion.

ber of non-zero coefficients (as in the case of l_0 pseudo norm that OMP uses) but also penalizes their magnitudes. This increases the temporal smoothness of the enhanced speech [15].

There are two problems with GDL. First, the two sub-dictionaries $\mathbf{D}_s$ and $\mathbf{D}_n$ are learnt independently – see Eqs. (11) and (12), and thus the source confusion error is not considered. Figure 1 illustrates speech and noise confusion. The second problem is the coherence between $\mathbf{D}_s$ and $\mathbf{D}_n$ that is also not considered in the learning process. These problems will be addressed in our DL algorithm proposed below.

3. IDDL Algorithm

The proposed dictionary learning algorithm defines a new cost function that penalizes coherence between the speech and the noise sub-dictionaries $\|\mathbf{D}_s^T\mathbf{D}_n\|_F^2$, source confusion $\|\mathbf{D}_n\mathbf{X}_s^n\|_F^2$ and noise confusion $\|\mathbf{D}_s\mathbf{X}_n^s\|_F^2$. The proposed algorithm iterates between three steps, after initializing the dictionary, sparse coding using LARC is performed – Eq. (13), then both $\mathbf{X}$ and the noise sub-dictionary $\mathbf{D}_n$ are fixed, while the speech sub-dictionary $\mathbf{D}_s$ is updated using Eq. (14), and in the third step both $\mathbf{X}$ and speech sub-dictionary $\mathbf{D}_s$ are fixed, while the noise sub-dictionary $\mathbf{D}_n$ is updated using Eq. (15).

1. Update $\mathbf{X}$ ($\mathbf{D}-[\mathbf{D}_s,\mathbf{D}_n]$ fixed):

$$\mathbf{X}=\min_{\mathbf{X}}\left\|\mathbf{Y}-\mathbf{D}\mathbf{X}\right\|_F^2+\lambda_l\left\|\mathbf{X}\right\|_1 \tag{13}$$

2. Update $\mathbf{D}_s$ ($\mathbf{X},\mathbf{D}_n$ fixed):

$$\mathbf{D}_s=\min_{\mathbf{D}_s}\left\|\mathbf{Y}_s-\mathbf{D}_s\mathbf{X}_s^s\right\|_F^2+\lambda_{nc}\left\|\mathbf{D}_sX_n^s\right\|_F^2+\lambda_c\left\|D_n^TD_s\right\|_F^2 \tag{14}$$

3. Update $\mathbf{D}_n$ ($\mathbf{X},\mathbf{D}_s$ fixed):

$$\mathbf{D}_n=\min_{\mathbf{D}_n}\left\|\mathbf{Y}_n-\mathbf{D}_n\mathbf{X}_n^n\right\|_F^2+\lambda_{sc}\left\|\mathbf{D}_nX_s^n\right\|_F^2+\lambda_c\left\|\mathbf{D}_s^T\mathbf{D}_n\right\|_F^2 \tag{15}$$

The λ_{nc}, λ_{sc}, $\lambda_c\geq 0$ are regularization parameters to control the importance of noise confusion, speech confusion, and sub-dictionary coherences, respectively.

We propose to use the alternating direction method of multipliers (ADMM) [35] to solve Eqs. (14) and (15). First, we introduce an auxiliary variable $\mathbf{Z}$ with an equality constrained $\mathbf{Z}=\mathbf{D}_s$:

$$\mathbf{D}_s=\arg\min_{\mathbf{D}_s,\mathbf{Z}}+\left\|\mathbf{Y}_s-\mathbf{D}_s\mathbf{X}_s^s\right\|_F^2+\lambda_{nc}\left\|\mathbf{D}_s\mathbf{X}_n^s\right\|_F^2+\lambda_c\left\|\mathbf{D}_n^T\mathbf{Z}\right\|_F^2,\quad \mathbf{Z}=\mathbf{D}_s\,. \tag{16}$$

We can see that Eq. (16), which is exactly equivalent to Eq. (14), is in the form that the ADMM algorithm solves. Next, we introduce the dual Lagrangian variable $\mathbf{U}$, and a scaling parameter p, to formulate the augmented Lagrangian, which is a function of the three variables, denoted as follows:

$$\mathcal{L}_\rho(\mathbf{D}_s,\mathbf{Z},\mathbf{U})=\left\|\mathbf{Y}_s-\mathbf{D}_s\mathbf{X}_s^s\right\|_F^2+\lambda_{nc}\left\|\mathbf{D}_s\mathbf{X}_n^s\right\|_F^2+\lambda_c\left\|\mathbf{D}_n^T\mathbf{Z}\right\|_F^2+\mathbf{U}(\mathbf{D}_s-\mathbf{Z})+\frac{\rho}{2}\left\|\mathbf{D}_s-\mathbf{Z}+\mathbf{U}\right\|_F^2\,. \tag{17}$$

According to ADMM, problem given by Eq. (16) can be solved by alternatingly updating, one at a time, each variable in Eq. (17), to minimize the augmented Lagrangian, while fixing the others. With an initial $\mathbf{Z}^0=\mathbf{U}^0=\mathbf{D}_s^0$ (the upper-script denotes the iteration time index), we alternatively solve the following problems until convergence is achieved:

$$\mathbf{D}_s^{t+1}=\min_{\mathbf{D}_s}\left\|\mathbf{Y}_s-\mathbf{D}_s\mathbf{X}_s^s\right\|_F^2+\lambda_{nc}\left\|\mathbf{D}_s\mathbf{X}_n^s\right\|_F^2+\lambda_c\left\|\mathbf{D}_n^T\mathbf{Z}^t\right\|_F^2+\mathbf{U}^t\left(\mathbf{D}_s-\mathbf{Z}^t\right)+\frac{\rho}{2}\left\|\mathbf{D}_s-\mathbf{Z}^t+\mathbf{U}^t\right\|_F^2\,, \tag{18}$$

$$\mathbf{Z}^{t+1}=\left(2\lambda_c\mathbf{D}_n\mathbf{D}_n^T+\rho\mathbf{I}\right)^{-1}\left(\mathbf{D}_s^{t+1}+\mathbf{U}^T\right)\,, \tag{19}$$

$$\mathbf{U}^{t+1}=\mathbf{U}^T+\rho\left(\mathbf{D}_s^{t+1}-\mathbf{Z}^{t+1}\right)\,. \tag{20}$$

After some matrix manipulation, we can easily see that Eq. (19) is equivalent to the following problem:

$$\mathbf{D}_s^{t+1}=\min_{\mathbf{D}_s}\mathrm{tr}\left(\mathbf{D}_s^T\mathbf{D}_s\mathbf{A}\right)-2\mathrm{tr}\left(\mathbf{D}_s^T\mathbf{B}\right)\,, \tag{21}$$

where

$$\mathbf{A}=\mathbf{Y}_s.\left(\mathbf{X}_s^s\right)^T+\frac{\rho}{2}\left(\mathbf{Z}^t-\mathbf{U}^t\right)\,,$$

$$\mathbf{B}=\mathbf{X}_s^s\left(\mathbf{X}_s^s\right)^T+\lambda_{nc}\mathbf{X}_n^s\left(\mathbf{X}_n^s\right)^T+\frac{\rho}{2}\mathbf{I}\,.$$

We can solve Eq. (21) using the same dictionary update algorithm as proposed by Mairal *et al.* [37] in online dictionary learning (ODL). This dictionary update algorithm is based on block-coordinate descent with warm start, which enjoys being parameter-free [37]. The same procedure procedure applies to the problem of learning the noise sub-dictionary $\mathbf{D}_n$ (Eq. (15)). Algorithm 1 describes the IDDL algorithm, while Algorithm 2 describes how sub-dictionary is updated using ADMM.

Algorithm 1: IDDL

Input: $\mathbf{Y}_s\in\mathbb{R}^{N\times n_s}$; $\mathbf{Y}_n\in\mathbb{R}^{N\times n_n}$; L; max_iter1; μ; λ_{nc}; λ_{sc}; λ_c
Output: $\mathbf{D}\in\mathbb{R}^{N\times 2L}$

Initialize $\mathbf{D}_s^0$, $\mathbf{D}_n^0$
$\mathbf{D}=[\mathbf{D}_s^0,\ \mathbf{D}_n^0]$
$\mathbf{Y}=[\mathbf{Y}_s,\ \mathbf{Y}_n]$
For $t=1$ to max_iter1 **do**
    $\mathbf{X}=\mathrm{LARC}(\mathbf{D},\mathbf{Y},\mu)$
    Update $\mathbf{D}_s$ using Algorithm 2, $i=1,\ j=2,\ \lambda=\lambda_{nc}$
    Update $\mathbf{D}_n$ using Algorithm 2, $i=2,\ j=1,\ \lambda=\lambda_{sc}$
    $\mathbf{D}=[\mathbf{D}_s,\mathbf{D}_n]$
End for

Algorithm 2: Sub-dictionary update

Input: $\mathbf{Y}_s; \mathbf{Y}_n; \mathbf{X}_s; \mathbf{X}_n; L; \mathbf{D} \in \mathbb{R}^{N\times 2L}; i; max\_iter2; \lambda; \lambda_c; \rho$
Output: $\mathbf{D}_i \in \mathbb{R}^{N\times L}$

$\mathbf{D}_i^0 = \mathbf{D}(:,(i-1).L+1:i.L)$
$\mathbf{D}_j = \mathbf{D}(:,(j-1).L+1:j.L)$
initial $\mathbf{Z}^0 = \mathbf{U}^0 = \mathbf{D}_i^0$
$\mathbf{X}_i^i = \mathbf{X}_i\big((i-1).L+1:i.L,:\big)$
$\mathbf{X}_j^i = \mathbf{X}_j\big((i-1).L+1:i.L,:\big)$
For $t = 1$ to max_iter 2 **do**
  Update $\mathbf{D}_i^t$ using Algorithm 2 in [37], where:
    $\mathbf{A} = \mathbf{Y}_i.(\mathbf{X}_i^i)^T + \frac{\rho}{2}(\mathbf{Z}^t - \mathbf{U}^t)$,
    $\mathbf{B} = \mathbf{Y}_i.(\mathbf{X}_i^i)^T + \lambda\mathbf{X}_j^i(\mathbf{X}_j^i)^T + \frac{\rho}{2}\mathbf{I}$
  $\mathbf{Z}^{t+1} = (2\lambda_c\mathbf{D}_j\mathbf{D}_j^T + \rho\mathbf{I})^{-1}(\mathbf{D}_i^{t+1} + \mathbf{U}^t)$
  $\mathbf{U}^{t+1} = \mathbf{U}^t + \rho(D_i^{t+1} - \mathbf{Z}^{t+1})$
End for

It should be noted that IDDL differs from FDDL in four aspects. First, we do not impose discrimination on the sparse codes $\mathbf{X}$ (i.e. $\lambda_2 = 0$). Second, the confusion error terms are weighted with regularization parameters. Third, a coherence penalty term is added to the DL formulation, and last, LARC is used at the sparse coding stage, instead of the fast iterative shrinkage and thresholding algorithm (FISTA) [36].

3.1. Speech Enhancement System based on DL

The overall speech enhancement system is depicted in Fig. 2. The system consists of two stages: training and enhancement. During the training phase, we learn the IDDL dictionary that models the amplitude spectrum of the training speech and noise samples. The amplitude of the short time Fourier coefficients (STFT) for the overlapping time frames of the clean speech and noise training signals is calculated after applying the Hamming window. The amplitude spectrum coefficients for all training frames are concatenated as columns to form $\mathbf{Y}_s$ and $\mathbf{Y}_n$, and fed to the IDDL algorithm that learns the clean speech sub-dictionary $\mathbf{D}_s$, and the noise sub-dictionary $\mathbf{D}_n$. These two sub-dictionaries are concatenated together to form the overall dictionary, that contains $2L$ columns.

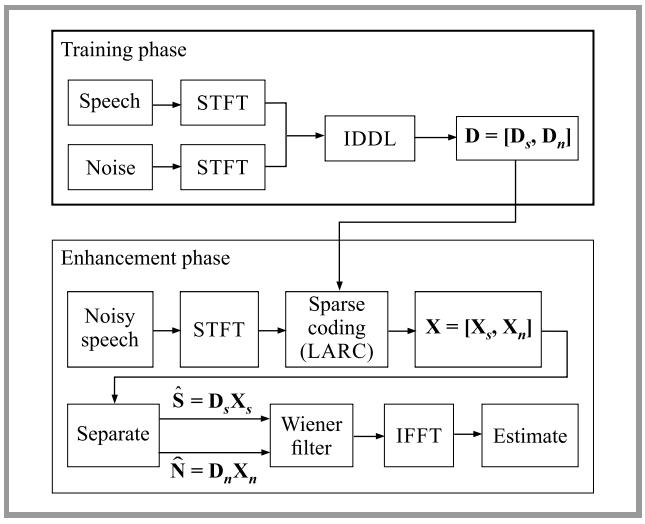

Fig. 2. The overall speech enhancement system.

At the enhancement phase, using LARC and the dictionary $\mathbf{D}$, the sparse codes $\mathbf{X}$ for the amplitude spectrum coefficients of the overlapping frames of the noisy signal are calculated. The sparse code vectors $\mathbf{X}$ contain $2L$ coefficients. The first L ones $\mathbf{X}_s$ that correspond to sub-dictionary $\mathbf{D}_s$ are separated from the last L ones $\mathbf{X}_n$ that correspond to sub-dictionary $\mathbf{D}_n$. By multiplying $\mathbf{X}_s$ and $\mathbf{D}_s$, as well as $\mathbf{X}_n$ and $\mathbf{D}_n$ we get an initial estimation for the amplitude spectrum of the clean speech and noise signals, respectively. These initial estimations are fed to the Wiener filter to find the final clean speech amplitude spectrum estimation. Finally, we apply the inverse Fourier transform to the estimated amplitude spectrum combined with the noisy phase spectrum to get the estimated clean speech.

4. Experiments

4.1. Noizeus Dataset

Noizeus [38] is a noise database that contains 30 IEEE sentences produced by three male and three female speakers, with 5 different sentences per speaker. The sentences are corrupted by eight different real-world noises at different SNRs: (0, 5, 10, 15) dB. The noise was taken from the Aurora database and includes suburban train noise, babble, car, exhibition hall, restaurant, street, airport and train station noise [38]. All speech and noise signals are sampled at 8 kHz.

As the database contains a small number of speakers, and to assure speaker independent cases, which means that the speakers in the training set are different from the speakers in the test set[2], we have divided the dataset into two sets: a training set that contains three speakers and another testing set that contains the remaining three speakers. We have created 12 training/test sets through permutations[3] of three speakers out of 6, and averaged the results. Table 1 shows the list of speakers and their characteristics [40], while Table 2 shows the speakers in the 12 different training/test sets we have created for experiments.

All training sets contain male and female speakers, except for the training set number 11, which is all female speakers (speakers 3, 4, and 6) while the corresponding test set is all male (speakers 1, 2, and 5), and the training set number 12 which is all male, while the corresponding test set is all female. The first 5 training set contain 2 male speakers and

[2] A speaker-independent scenario enables the proposed system to use any available clean speech samples as the training set, not necessarily pertaining to the speaker whose speech we wish to enhance, contrary to the speaker-dependent scenario.

[3] Permutation of 3 speakers out of 6 gives 20 sets. For our experiments we took only 10 sets that contain both male and female speakers, to calculate the average PESQ and fwSegSNR. We conducted the experiment using set number 11 (see Table 2) that contains a female speaker only, and set number 12 that contains a male speaker only, to see if the gender of the speakers in the training set has any impact on the performance achieved. The average does not include results for test groups 11 and 12.

Table 1
Noizeus speaker's characteristics

Speaker	Gender	Pitch frequency F0 [Hz]	Age	State raised	Sentences
1	M	135	21	Texas	sp1–sp5
2	M	134	20	California	sp6–sp10
3	F	225	19	North Carolina and Texas	sp11–sp15
4	F	245	22	Texas	sp16–sp20
5	M	144	20	Texas and Kentucky	sp21–sp25
6	F	225	22	Texas	sp26–sp30

Table 2
Description of different training and test sets

Train/test set	Speakers in the training set	Speakers in the test set
1	[1, 2, 3]	[4, 5, 6]
2	[1, 2, 4]	[3, 5, 6]
3	[1, 2, 6]	[3, 4, 5]
4	[2, 3, 5]	[1, 4, 6]
5	[2, 4, 5]	[1, 3, 6]
6	[1, 3, 4]	[2, 5, 6]
7	[2, 3, 4]	[1, 5, 6]
8	[3, 4, 5]	[1, 4, 5]
9	[3, 4, 6]	[1, 2, 6]
10	[4, 5, 6]	[1, 2, 3]
11	[3, 4, 6]	[1, 2, 5]
12	[1, 2, 5]	[3, 4, 6]

1 female, while the other 5 training sets contain 2 female speakers and 1 male.

For every training set, we collect 15 sentences (15 clean recordings and 15 noisy recordings of a specific type of noise) uttered by the 3 chosen speakers. To have the noise training samples (that belong to the specific type, e.g. a car), we subtract the clean speech recordings from the noisy recordings within the training dataset, and thus we get 15 recordings per noise[4]. Every recording (clean speech and noise) is divided into overlapping frames that are 128 ms long, with a ratio of 75% (overlapping length of 96 ms). After applying the hamming window to these clean speech and noise frames, we calculate the FFT coefficients of these overlapped windows, and stack them together as columnar vectors to form the training matrices $\mathbf{Y}_s$ (the amplitude of the FFT coefficients of the clean speech frames) and $\mathbf{Y}_n$ (the amplitude of the FFT coefficients of the clean speech frames). The same procedure is applied to noisy signals at the enhancement stage.

[4] In the general case, noise samples can be obtained either through a voice activity detector (VAD) from non-speech segments, or from an offline noise database like Noisex-92 [43].

4.2. Performance Metrics

There are two types of measures to assess the performance of speech enhancement algorithms: subjective measures and objective measures. Subjective measures are scores reported by human listeners participating in a subjective listening test. The mean opinion score (MOS) [40] is a result, on the scale from 1 to 5, that a human listener decides to use to express their satisfaction with the quality of speech they are listening to. Due to the high logistic costs needed to perform subjective listening tests, objective measures were sought.

Frequency weighted segmental SNR (fwSegSNR) is the estimated mean frequency domain SNR over all time frames, with a perceptually motivated frequency band weighting. fwSegSNR may be calculated according as follows:

$$\text{fwSegSNR} = \frac{10}{Nw}\sum_{n=1}^{N}\sum_{b=1}^{B} w_b \log \frac{|S(b,n)|^2}{(|S(b,n)| - |\hat{S}(b,n|)^2}\,, \tag{22}$$

where $S(b,n)$ are the complex FFT coefficients of the clean speech, n is the frame index, b is the frequency component index, N is the total number of frames in the speech signal, B is the total number of frequency components, w_b is the corresponding frequency weighting, w is the sum of all the frequency weights, and $\hat{S}(b,n)$ are the estimated complex spectrum coefficients of the enhanced speech.

PESQ [39] is an international measure that simulates MOS, and is widely used to assess the quality of speech conveyed through a telephone network. Its derivation may be found in [39]. It has been shown that PESQ has the highest correlation coefficient ($\rho = 0.89$) with the overall speech quality [41], and correlates well with subjective speech *intelligibility* [42], while fwSegSNR has the second highest correlation coefficients with the overall speech quality ($\rho = 0.85$) [41], and correlates well with subjective speech *quality* [42]. That is why we have used these two measures to assess performance of the proposed algorithm. It should be emphasized that speech quality does not correlate with speech intelligibility, as it is the case of synthesized speech, which generally has low quality, though it could be highly intelligible. We have used the implementation provided by [40] for both fwSegSNR and PESQ.

4.3. The Results

To assess the performance of the proposed IDDL algorithm, we compared its performance in terms of fwSegSNR and PESQ against three other different DL algorithms: K-SVD, GDL, and FDDL. We have used the same speech enhancement system as depicted in Fig. 2, but with the different DL algorithms mentioned.

Different frame lengths were investigated starting with 256 up to 1024 samples (from 32 to 128 ms) with the overlapping rate of 50–75%. We have found that longer frames always render better results. This increases the dimensions of the feature space and, thus, results in a lower coherence between the clean speech and noise sub-dictionaries, which

means a lower source confusion error. Longer enhancement time is a disadvantage of using the longer frame length, as increasing the dimensions of the signal feature space N will increase the time needed for sparse coding.

The number of DFT coefficients varied as well. In one scenario we chose the number of DFT coefficients as the same number of samples in the frame, while for short frame lengths we tried zero padding and used 1024 DFT coefficients, but it was not useful. This is because it does not really increase the dimension of the feature space, as the information content is the same.

The regularization parameters λ_{nc}, λ_{sc}, λ_c are set through a validation test. We found that the optimal experimental values in our setting are: $\lambda_{nc} = 1$, $\lambda_{sc} = 1$, $\lambda_c = 0.0001$.

For LARC, the stopping residual coherence thresholds is set at $\mu = 0.15$ in IDDL and GDL at the training stage, while it's set at $\mu = 0.1$ for sparse coding at the enhancement stage, as described in [15]. We have verified experimentally that those values of μ are optimal for the performance of the proposed system, for all noise types except for the case of white noise and car noise. We have found that using a lower value of $\mu = 0.05$ for sparse coding renders better performance for all dictionaries studied, in the case of car noise and white noise. This is due to the fact that both types of noise experience lower confusion levels (non-speech-like noise) with clean speech signal than other types of speech-like noise, and using a lower value of μ (which means a lower approximation error) will not cause the confusion error to increase. This hints that we can use a dynamic value for μ based on the initial value of the speech confusion error.

For FDDL, we have used the efficient implementation provided by [31] and denoted by E-FDDL. There are two parameters to tune E-FDDL: FISTA l_1 regularization parameter which is set to $\lambda_1 = 0.05$, and Fisher discrimination regularization parameter $\lambda_2 = 0.01$, Eq. (6).

The number of maximum iterations for KSVD is set to 15, for E-FDDL it is set to 7, and for GDL it is set to 20.

The number of maximum iterations (max_iter1 in Algorithm 1) for IDDL is set to 7. In fact we have tested different values in the range of $\{3 \ldots 25\}$ for this parameter. We noticed that increasing the number of maximum iterations over 7 minimizes all sub-costs: source distortion, noise distortion, source confusion, noise confusion and sub-dictionaries coherence even further (see Figs. 6–10), but the resulting dictionary does not perform better on the testing set. This is due to the fact that the model (the dictionary) becomes over-fitted to the training set and does not generalize well.

For initializing IDDL ($\mathbf{D}_s^0, \mathbf{D}_n^0$), we have investigated two scenarios: either to build two initial dictionaries composed of random samples from the training set, or to initialize IDDL with two prebuilt K-SVD dictionaries, one for clean speech and one for noise. We have noticed that IDDL with a random initial dictionary has no performance gain over the other studied DL methods, while IDDL with initial prebuilt K-SVD dictionaries achieves superior performance.

Table 3
The proposed speech enhancement system's general parameter setting

Parameter	Variable and value
Window length	$Tw = 128$ ms
Window shift	$Tsh = 32$ ms
Number of FFT coefficients	$NFFT = 1024$
Signal dimension	$N = \frac{NFFT}{2} + 1 = 513$
Number of atom in the sub-dictionary	$L = 300$
The stopping residual coherence thresholds of LARC (the sparse coding block in the enhancement stage)	$\mu = 0.14$

Table 4
The parameters setting of the 4 used dictionary learning algorithms used in the training stage

Dictionary	Parameter	Variable name
IDDL	The stopping residual coherence thresholds of LARC (sparse coding stage): lines 3, 4 in Algorithm 1	$\mu = 0.15$
	Distortion error penalty	$\lambda_{nc} = 1$, $\lambda_{sc} = 1$
	Coherence penalty	$\lambda_c = 0.0001$
	max_iter1	$q_1 = 7$
	max_iter2	$q_2 = 5$
FDDL	FISTA $l1$ regularization	$\lambda_1 = 0.05$
	Fisher discrimination regularization parameter	$\lambda_2 = 0.01$
	Maximum iterations	$q = 7$
K-SVD	Sparsity degree (for OMP stage)	$k = 30$
	Maximum iterations	$q = 15$
GDL	The stopping residual coherence thresholds of LARC (sparse coding solver)	$\mu = 0.15$
	Maximum iterations	$q = 20$

The results shown in Tables 4 and 5 are for the case $L = 300$ (which means that the dimensions of the total dictionary are $513^7 \times 600$), with a pre-built K-SVD dictionary for initialization.

Experiments were conducted using Matlab 2015Ra on a laptop with 3.16 GHz Intel Core i5 processor and 4 GB RAM.

Table 3 summarizes the general parameters settings of the speech enhancement system, while Table 4 summarizes all dictionary learning parameters, for the 4 dictionary learning algorithms used to get the results reported in Tables 4 and 5.

Table 5 shows the frequency weighted segmental SNR (in dB) for the different dictionary learning algorithms using the parameter settings listed above in Tables 3 and 4. To

Table 5
Frequency weighted segmental SNR (in dB), speaker independent scenario, with the percentage gain of IDDL and FDDL gain over K-SVD and GDL

Noise	dB	K-SVD	GDL	FDDL	IDDL	IDDL gain [%]	FDDL gain [%]
Babble	0	6.17	6.13	6.23	**6.25**	**1.30**	0.96
	5	7.89	7.97	7.91	**7.99**	**0.25**	–0.75
	10	9.76	9.92	9.70	**9.93**	**0.10**	–2.26
	15	11.90	12.18	12.13	**12.25**	**0.57**	–0.41
Car	0	7.14	7.16	7.53	**7.60**	**6.15**	4.91
	5	8.55	8.58	8.86	**9.11**	**6.18**	3.16
	10	10.26	10.35	10.63	**10.73**	**3.67**	2.63
	15	12.57	12.69	12.76	**12.77**	**0.63**	0.54
Restaurant	0	6.52	6.48	6.54	**6.55**	**0.46**	0.30
	5	7.83	7.95	7.86	**7.96**	**0.13**	–1.14
	10	9.66	9.75	9.49	**9.77**	**0.21**	–2.73
	15	11.44	11.64	11.46	**11.85**	**1.80**	–1.57
Station	0	6.06	6.10	**6.25**	6.17	1.15	**2.40**
	5	7.91	8.04	**8.07**	8.03	–0.12	**0.37**
	10	9.88	10.02	9.96	**10.03**	**0.10**	–0.60
	15	11.96	12.09	11.95	**12.19**	**0.83**	–1.17
Train	0	7.57	7.50	7.70	**7.85**	**3.70**	1.68
	5	8.91	8.56	8.96	**9.02**	**1.23**	0.55
	10	10.30	10.46	10.74	**10.82**	**3.44**	2.60
Airport	0	6.65	6.54	6.74	6.73	1.20	**1.33**
	5	8.16	8.28	8.30	8.30	0.24	0.24
	10	10.22	10.19	10.13	**10.24**	**0.20**	–0.88
	15	12.25	12.31	12.29	**12.42**	**0.89**	–0.16
White	0	**7.11**	6.98	6.92	6.97	–1.97	–2.74
	5	**8.68**	8.49	8.44	8.54	–1.61	–2.84
	10	**10.56**	10.26	10.28	10.40	–1.52	–2.72
	15	**12.73**	12.28	12.63	12.41	–2.51	–0.79

Table 6
PESQ, speaker independent scenario, with the percentage gain of IDDL and FDDL over K-SVD and GDL

Noise	dB	K-SVD	GDL	FDDL	IDDL	IDDL gain [%]	FDDL gain [%]
Babble	0	1.87	1.89	1.94	**1.95**	**3.17**	2.57
	5	2.19	2.20	2.23	**2.23**	**1.36**	1.34
	10	2.46	2.51	**2.52**	2.51	0.00	**0.39**
	15	2.76	2.85	**2.85**	2.82	–1.05	0
Car	0	2.24	2.28	2.36	**2.40**	**5.26**	3.38
	5	2.43	2.49	2.55	**2.58**	**3.61**	2.35
	10	2.61	2.68	2.71	**2.75**	**2.61**	1.10
	15	2.82	2.93	2.93	**2.95**	**0.68**	0
Restaurant	0	1.87	1.88	1.91	**1.92**	**2.13**	1.57
	5	2.11	2.13	2.17	**2.17**	**1.88**	1.84
	10	2.44	2.47	2.49	**2.49**	**0.81**	0.80
	15	2.68	**2.78**	2.77	2.75	–1.08	**0.35**
Station	0	1.89	1.94	**1.98**	1.97	1.55	**2.02**
	5	2.23	2.29	**2.33**	2.32	1.31	**1.71**
	10	2.50	2.57	**2.59**	2.58	0.39	**0.77**
	15	2.74	2.81	**2.82**	2.80	–0.36	**0.35**
Train	0	2.32	2.23	**2.40**	2.37	2.16	**3.33**
	5	2.46	2.40	**2.55**	2.53	2.85	**3.52**
	10	2.52	2.61	**2.74**	2.73	4.60	**4.74**
Airport	0	1.94	1.93	**1.99**	1.97	1.55	**2.51**
	5	2.25	2.26	**2.30**	2.29	1.33	**1.73**
	10	2.52	2.53	**2.57**	2.55	0.79	**1.55**
	15	2.79	2.81	**2.85**	2.82	0.36	**1.40**
White	0	**2.39**	2.32	2.38	2.39	0.00	–0.42
	5	**2.63**	2.54	2.61	2.63	0.00	–0.76
	10	**2.84**	2.75	2.83	2.83	–0.35	–0.35
	15	**3.03**	2.95	3.03	3.02	–0.33	0

evaluate the degree of improvement that IDDL and FDDL (which are discriminative dictionary learning algorithms) offer over K-SVD and GDL (which are reconstructive DL), we reported, in the same table, the percentage gain of IDDL and FDDL over K-SVD and GDL (percentage of outperformance), which is calculated as:

$$IDDL_{GAIN}=\left(1-\frac{\left|\max\left(\text{fwSegSNR(KSVD)},\text{fwSegSNR(GDL)}\right)\right|}{\text{fwSegSNR(IDDL)}}\right)\cdot 100\%, \tag{23}$$

$$FDDL_{GAIN}=\left(1-\frac{\left|\max\left(\text{fwSegSNR(KSVD)},\text{fwSegSNR(GDL)}\right)\right|}{\text{fwSegSNR(FDDL)}}\right)\cdot 100\%. \tag{24}$$

We can see that the proposed IDDL algorithm performs better in terms of fwSegSNR in most cases (19 out of 27), but not in the case of white noise, as it is not a structured noise.

Table 6 shows PESQ for the different dictionary learning algorithms, with the IDDL and FDDL percentage gain over K-SVD and GDL, which is calculated from Eqs. (23) and (24), using PESQ instead of fwSegSNR. The results show that IDDL performs better in 9 out of 27 of the cases, while its performance is very close to that of E-FDDL in the remaining cases.

Tables 5 and 6 show that K-SVD is the best DL for the case of white noise, in terms of both performance measures, and no gain is achieved by IDDL nor FDDL.

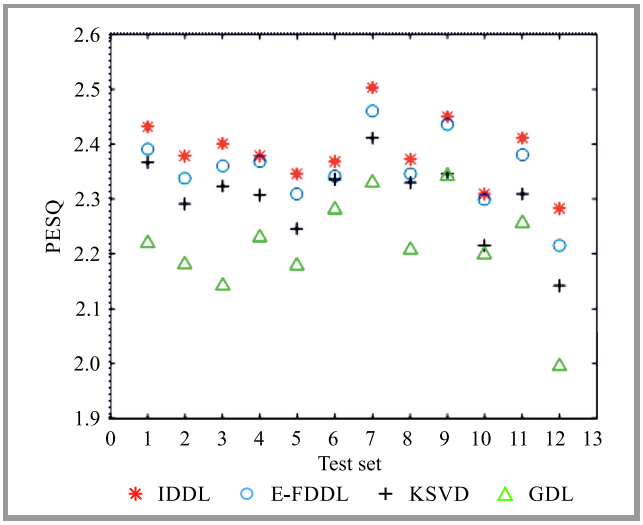

Fig. 3. PESQ over the different test sets.

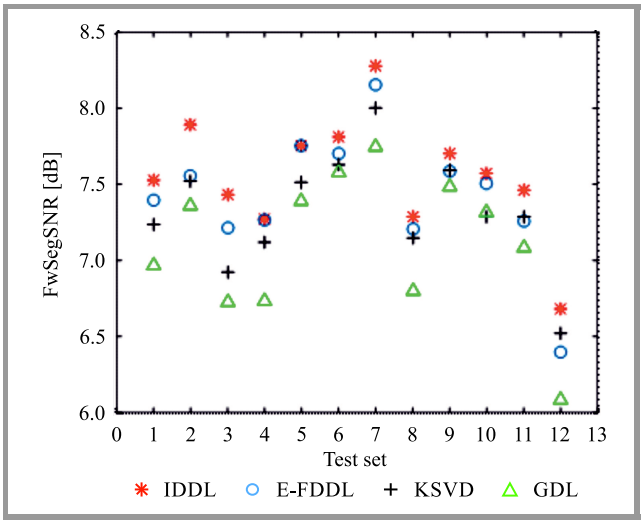

Fig. 4. fwSegSNR over the different test sets.

Figures 3 and 4 show fwSegSNR and PESQ, respectively, over all different test sets, in the case of car noise at 0 dB. We can see that IDDL outperforms all other investigated DL algorithms, over all test sets. We also noticed that the worst performance (lowest fwSegSNR and lowest PESQ) for all dictionaries is when the training set is all male, while the testing set is all female (test set number 12), while when the training set is all female and the testing set is all male (test set number 11), performance does not degrade, which might hint that the model learnt from female voices generalizes better than the model learnt from male voices (which needs to be investigated further in the future).

Table 7
DL training time in seconds

Number of atoms L	KSVD	GDL	FDDL	IDDL
300	**15**	44	52	20
600	**41**	58	175	57

Table 7 shows the different DL training times. We can see that K-SVD is characterized by the shortest DL time, while IDDL is ranked second. FDDL has the longest training time, because at the sparse code updating stage it enforces discrimination using Fisher discrimination criteria on the sparse codes, which is a costly sparse coding algorithm. Although Tables 5 and 6 show that IDDL offer performance that is very close to that of E-FDDL in terms of both performance measures, it has the advantage of lower complexity, and thus a short training time.

Table 8
Sparse coding time

Number of atoms L	Coding time [s]
300	0.008
600	0.03

Table 8 shows the different coding times for a single noisy frame using LARC, for different dictionary sizes. As expected, we notice that increasing the dictionary size (increasing the number of atoms L) increases the time needed to calculate the sparse codes $\mathbf{x}_i$ (which has a dimension of $2L$) of the amplitude spectrum of each noisy frame, at the enhancement stage, and thus increases the time needed to perform speech enhancement.

4.4. Convergence Analysis

We have studied empirically the convergence of the proposed dictionary learning through examining how all the IDDL sub-costs (speech distortion, noise distortion, speech confusion, noise confusion, and sub-dictionaries' coherence) change with the respective iterations (variable t in Algorithm 1, line 2). All reported figures relate to babble noise, with 0 dB. Figure 5 shows that speech and noise distortion decreases with the number of iterations. We can also see that speech distortion is smaller than

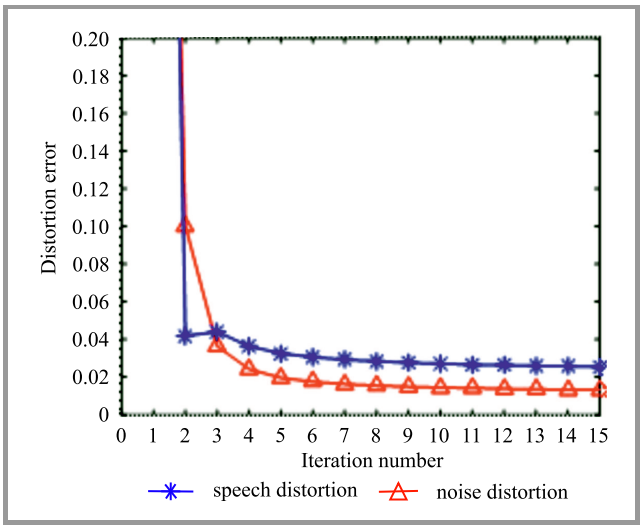

Fig. 5. Speech and noise distortion.

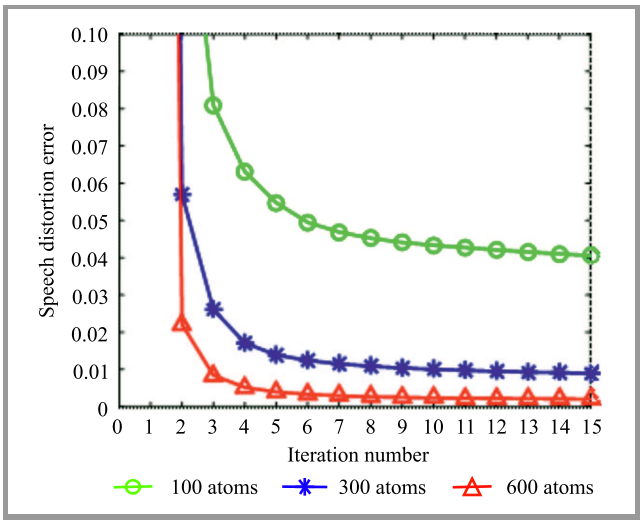

Fig. 6. Speech distortion for different dictionary sizes.

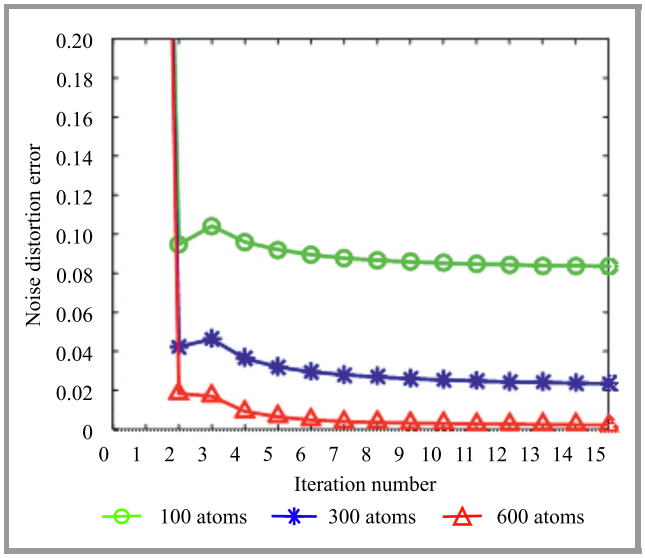

Fig. 7. Noise distortion for different dictionary sizes.

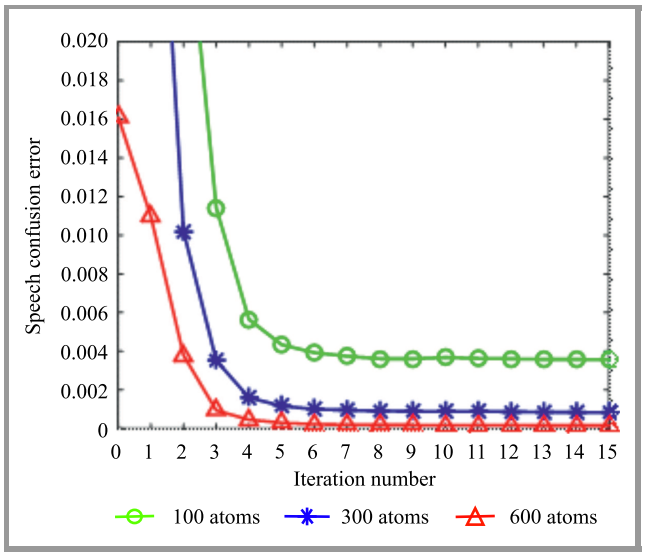

Fig. 8. Speech confusion error for different dictionary sizes.

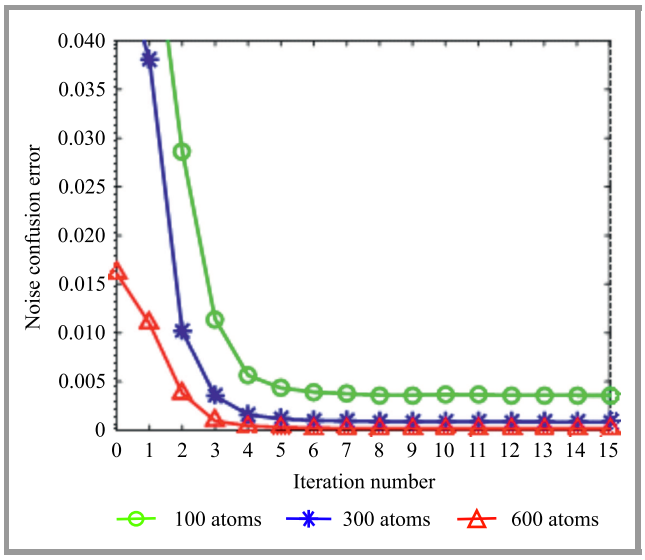

Fig. 9. Noise confusion error for different dictionary sizes.

noise distortion. This is because speech is more structured than noise.

Figures 6 and 7 show speech distortion and noise distortion for different number of atoms, respectively, and we can see that increasing the number of atoms decreases the distortion error, as the dictionary becomes richer, and thus has higher representation capability.

Figures 8 and 9 show the speech confusion error and noise confusion error. We can see that both speech and noise confusion errors achieve a considerable decrease with iteration number 3.

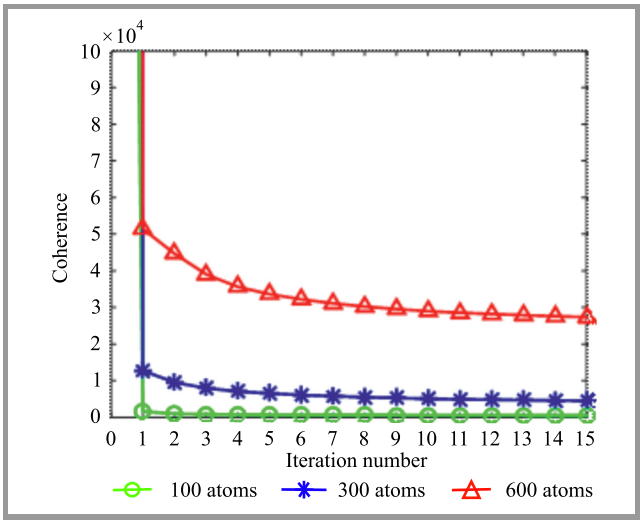

Fig. 10. Coherence between speech and noise sub-dictionaries.

Figure 10 shows the coherence between the noise and speech sub-dictionaries. We can see that increasing the number of atoms increases the coherence, as the minimum coherence increases with increasing the number of columns in any matrix.

5. Conclusion

In this paper we proposed a new algorithm to learn an incoherent discriminative dictionary called IDDL, used for the specific task of speech enhancement. The goal of the cost function is to minimize both "source distortion" and "source confusion" errors, in addition to reducing coherence between noise and speech sub-dictionaries. Performance of the proposed algorithm was evaluated using two objective measures: frequency weighted SNR: fwSegSNR and PESQ, to compare with well-known dictionary learning algorithms: K-SVD, GDL and FDDL. Experiments on the Noizeus dataset show that IDDL offers better performance in comparison to other studied DL in terms of both measures, in most of the cases, but not in the case of white noise. Performance of IDDL is close to that of E- FDDL in terms of both performance measures, but it has the advantage of having a notably shorter training time. The superior performance of IDDL makes it suitable for speech enhancement in the case of structured non-stationary noise, such as babble and car noise.

Acknowledgments

The authors would like to thank P. Loizou (R.I.P.) for publishing the implementations of fwSegSNR and PESQ objective measures. We thank Christian D. Sigg for providing an excellent implementation of LARC algorithm and GDL. We also thank Tiep Huu Vu for providing an excellent and efficient implementation of E-FDDL.

References

[1] S. Boll, "Suppression of acoustic noise in speech using spectral subtraction", *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 27, no. 2, pp. 113–120, 1979 (doi: 10.1109/TASSP.1979.1163209).

[2] Y. Lu and P. C. Loizou, "A geometric approach to spectral subtraction", *Speech Commun.*, vol. 50, no 6. pp. 453–466, 2008 (doi: 10.1016/j.specom.2008.01.003).

[3] J. S. Lim and A. V. Oppenheim, "Enhancement and bandwidth compression of noisy speech", *Proc. of the IEEE*, vol. 67, no. 12, pp. 1586–1604, 1979 (doi: 10.1109/PROC.1979.11540).

[4] Y. Ephraim, "Statistical-model-based speech enhancement systems", *Proc. of the IEEE*, vol. 80, no. 10, pp. 1526–1555, 1992 (doi: 10.1109./5.168664).

[5] Y. Hu and P. C. Loizou, "A generalized subspace approach for enhancing speech corrupted by colored noise", *IEEE Trans. Speech Audio Process.*, vol. 11, no. 4, pp. 334–341, 2003 (doi: 10.1109/TSA.2003.814458).

[6] J. Sun, J. Zhang, and M. Small, "Extension of the local subspace method to enhancement of speech with colored noise", *Signal Process.*, vol. 88, no. 7, pp. 1881–1888, 2008 (doi: 10.1016/j.sigpro.2008.01.008).

[7] T. Sreenivas and P. Kirnapure, "Codebook constrained Wiener filtering for speech enhancement", *IEEE Trans. Speech Audio Process.*, vol. 4, no. 5, pp. 383–389, 1996 (doi: 10.1109/89.536932).

[8] S. Srinivasan, J. Samuelsson, and W. B. Kleijn, "Codebook driven short term predictor parameter estimation for speech enhancement", *IEEE Trans. Audio, Speech, and Language Process.*, vol. 14, no. 1, pp. 163–176, 2006 (doi: 10.1109/TSA.2005.854113).

[9] D. Y. Zhao and W. B. Kleijn, "HMM-based gain modeling for enhancement of speech in noise", *IEEE Trans. Audio, Speech, and Language Process.*, vol. 15, no. 3, pp. 882–892, 2007 (doi: 10.1109/TASL.2006.885256).

[10] N. Mohammadiha, R. Martin, and A. Leijon, "Spectral domain speech enhancement using HMM state-dependent super-Gaussian priors", *IEEE Signal Process. Lett.*, vol. 20, no. 3, pp. 253–256, 2013 (doi: 10.1109/LSP.2013.2242467).

[11] H. Veisi and H. Sameti, "Speech enhancement using hidden Markov models in Mel-frequency domain", *Speech Commun.*, vol. 55, no. 2, pp. 205–220, 2013 (doi: 10.1016/j.specom.2012.2242467).

[12] K. W. Wilson, B. Raj, and P. Smaragdis, "Regularized non-negative matrix factorization with temporal dependencies for speech denoising", in *Proc. of the 9th Ann. Conf. of the Int. Speech Commun. Association, Brisbane Interspeech 2008*, Brisbane, Australia, 2008, pp. 411–414.

[13] M. Sun, Y. Li, J. Gemmeke, and X. Zhang, "Speech enhancement under low SNR conditions via noise estimation using sparse and low-rank NMF with Kullback-Leibler divergence", *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 23, no. 7, pp. 1233–1242, 2015 (doi: 10.1109.TASLP.2015.2427520).

[14] N. Mohammadiha, P. Smaragdis, and A. Leijon, "Supervised and unsupervised speech enhancement using nonnegative matrix factorization", *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 21, no. 10, pp. 2140–2151, 2013 (doi: 10.1109.TASL.2013.2270369).

[15] C. D. Sigg, T. Dikk, and J. M. Buhmann, "Speech enhancement using generative dictionary learning", *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 6, pp. 1698–1712, 2012 (doi: 10.1109.TASL.2012.2187194).

[16] Y. Zhao, X. Zhao, and B. Wang, "A speech enhancement method based on sparse reconstruction of power spectral density", *Computers & Elec. Engin.*, vol. 40, no. 4, 2014, pp. 1080–1089 (doi: 10.1016/j.compeleceng.2013.12.007).

[17] Y. Luo, G. Bao, Y. Xu, and Z. Ye, "Supervised monaural speech enhancement using complementary joint sparse representations", *IEEE Signal Process. Lett.*, vol. 23, no. 2, pp. 237–241, 2016 (doi: 10.1109/LSP.2015.2509480).

[18] L. Zhang, G. Bao, J. Zhang, and Z. Ye, "Supervised single-channel speech enhancement using ratio mask with joint dictionary learning", *Speech Commun.*, vol. 82, no. C, pp. 38–52, 2016 (doi: 10.1016/j.specom.2016.06.001).

[19] T. W. Shen and D. P K Lun, "A speech enhancement method based on sparse reconstruction on log-spectra", *HKIE Trans.*, vol. 24, no. 1, pp. 24–34, 2017 (doi: 10.1080/1023697X.2016.1210545).

[20] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries", *IEEE Trans. Image Process.*, vol. 15, no. 12, pp. 3736–3745, 2006 (doi: 10.1109/TIP.2006.881969).

[21] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation", *IEEE Trans. on Signal Process.*, vol. 54, no. 11, pp. 4311–4322, 2006 (doi: 10.1109/TSP.2006.881199).

[22] R. Rubinstein, M. Zibulevsky, and M. Elad, "Efficient implementation of the K-SVD algorithm using batch orthogonal matching pursuit", Technical Rep. CS-2008-08, Technion – Israel Institute of Technology, Haifa, Israel, 2008.

[23] K. Engan, S. O. Aase, and J. Hakon Husoy, "Method of optimal directions for frame design", in *Proc. IEEE Int. Conf. on Acoust., Speech, and Sig. Process. ICASSP'99*, Phoenix, AZ, USA, 1999, vol. 5 (doi: 10.1109/ICASSP.1999.760624).

[24] Y. Suo, M. Dao, U. Srinivas, V. Monga, and T. D. Tran, "Structured Dictionary Learning for Classification", 2014, arXiv:1406.1943.

[25] J. Mairal, F. Bach, J. Ponce, G. Sapiro, A. Zisserman, "Supervised dictionary learning", in *Advances in Neural Information Processing Systems 21*, D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, Eds. MIT Press, 2008, pp. 1033–1040.

[26] Q. Zhang and B. Li, "Discriminative K-SVD for dictionary learning in face recognition", in *Proc. 23rd IEEE Conf. on Comp. Vision and Pattern Recogn. CVPR 2010*, San Francisco, CA, USA, 2010, pp. 2691–2698 (doi: 10.1109/CVPR.2010.5539989).

[27] Z. Jiang, Z. Lin, and L. S. Davis, "Label consistent K-SVD: Learning a discriminative dictionary for recognition", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2651–2664, 2013 (doi: 10.1109/TPAMI.2013.88).

[28] I. Ramirez, P. Sprechmann, and G. Sapiro, "Classification and clustering via dictionary learning with structured incoherence and shared features", in *Proc. 23rd IEEE Conf. on Comp. Vision and Pattern Recogn. CVPR 2010*, San Francisco, CA, USA, 2010, pp. 3501–3508 (doi: 10.1109/CVPR.2010.5539964).

[29] S. Kong and D. Wang, "A dictionary learning approach for classification: Separating the particularity and the commonality", in *Proc. 12th Eur. Conf. on Comp. Vision ECCV 2012*, Florence, Italy, 2012, pp. 186–199 (doi: 10.1007/978-3-642-33718-5_14).

[30] M. Yang, L. Zhang, and X. Feng, "Sparse representation based Fisher discrimination dictionary learning for image classification", *Int. J. of Computer Vision*, vol. 109, no. 3, pp. 209–232, 2014 (doi: 10.1007/s11263-014-0722-8).

[31] T. H. Vu and V. Monga, "Fast low-rank shared dictionary learning for image classification", in *IEEE Trans. on Image Process.*, vol. 26, no. 11, pp. 5160–5175, 2017 (doi: 10.1109/TIP.2017.2729885).

[32] J. A. Tropp, "Greed is good: algorithmic results for sparse approximation", *IEEE Trans. on Inform. Theory*, vol. 50, no. 10, pp. 2231–2242, 2004 (doi: 10.1109/TIT.2004.834793).

[33] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition", in *Proc. of 27th Asilomar Conf. on Signals, Systems and Computers*, Pacific Grove, CA, USA, 1993 (doi: 10.1109/ACSSC.1993.342465).

[34] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression", *The Annals of Statistics*, vol. 32, no. 2, pp. 407–499, 2004 (doi: 10.1214/009053604000000067).

[35] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers", *Found. and Trends in Machine Learn.*, vol. 3, no. 1, pp. 1–122, 2011 (doi: 10.1561/2200000016).

[36] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems", *SIAM J. on Imaging Sci.*, vol. 2, no. 1, pp. 183–202, 2009 (doi: 10.1137/080716542).

[37] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding", *J. of Machine Learn. Res.*, vol. 11, pp. 19–60, 2010, arXiv:0908.0050.

[38] "Noizeus: A noisy speech corpus for evaluation of speech enhancement algorithms" [Online]. Available: http://ecs.utdallas.edu/loizou/speech/noizeus/

[39] A. Rix, J. Beerends, M. Hollier, and A. Hekstra, "Perceptual evaluation of speech quality (PESQ) – A new method for speech quality assessment of telephone networks and codecs", in *Proc. IEEE Int. Conf. on Acoust., Speech, and Sig. Process.*, Salt Lake City, UT, USA, 2001, pp. 749–752 (doi: 10.1109/ICASSP.2001.941023).

[40] P. C. Loizou, *Speech Enhancement. Theory and Practice*. Boca Raton, FL, USA: CRC, 2013 (ISBN: 9781138075573).

[41] Y. Hu and P. C. Loizou, "Evaluation of objective quality measures for speech enhancement", *IEEE Trans. Audio Speech Lang. Process.*, vol. 16, no. 1, pp. 229–38, 2008 (doi: 10.1109/TASL.2007.911054).

[42] J. Ma, Y. Hu, and P. C. Loizou, "Objective measures for predicting speech intelligibility in noisy conditions based on new band importance functions", *J. Acoust. Soc. Am.*, vol. 125, no. 5, pp. 3387–3405, 2009.

[43] Noisex-92: Database of recording of various noises [Online]. Available: www.speech.cs.cmu.edu/comp.speech/Section1/Data/noisex.html

Dima Shaheen is a Ph.D. student at the Higher Institute for Applied Science and Technology HIAST, Damascus, Syria. She received her M.Sc. degree in Telecommunication Engineering in 2013 from HIAST. Her research interests include signal processing, speech processing, machine learning, information theory, and digital communication.

E-mail: dima.shaheen@hiast.edu.sy
Telecommunications Department
Higher Institute for Applied Science
and Technology HIAST
P.O. Box 31983 Damascus, Syria

Oumayma Al Dakkak received her Ph.D. in Electronic Systems from the Grenoble Institute of Technology (Institut polytechnique de Grenoble) in 1988. She is a Research Director at the Telecommunications Department, Higher Institute for Applied Science and Technology HIAST, Damascus, Syria. Her research interests include signal processing, speech processing, speech recognition, machine learning, and digital communication.
E-mail: oumayma.dakkak@hiast.edu.sy
Telecommunications Department
Higher Institute for Applied Science
and Technology HIAST
P.O. Box 31983 Damascus, Syria

Mohiedin Wainakh received his Ph.D. degree in Cybernetics and Information Theory in 1980 from the Kiev Polytechnic, Ukraine. Currently, he is a Research Director at the Telecommunications Department, Higher Institute for Applied Science and Technology HIAST, Damascus, Syria. His current research interests include digital communication, statistics, and information theory.
E-mail: mohiedin.wainakh@hiast.edu.sy
Telecommunication Department
Higher Institute for Applied Science
and Technology HIAST
P.O. Box 31983 Damascus, Syria