A speech enhancement algorithm based on a non-negative hidden Markov model and Kullback-Leibler divergence

Xiang, Yang; Shi, Liming; Højvang, Jesper Lisby; Rasmussen, Morten Højfeldt; Christensen, Mads Græsbøll

Published in:
Eurasip Journal on Audio, Speech, and Music Processing

DOI (link to publication from Publisher):
10.1186/s13636-022-00256-5

Publication date:
2022

Document Version
Publisher's PDF, also known as Version of record

EURASIP Journal on Audio, Speech, and Music Processing

METHODOLOGY

Open Access

A speech enhancement algorithm based on a non-negative hidden Markov model and Kullback-Leibler divergence

Yang Xiang[1,2*], Liming Shi[1], Jesper Lisby Højvang[2], Morten Højfeldt Rasmussen[2] and Mads Græsbøll Christensen[1]

Abstract

In this paper, we propose a supervised single-channel speech enhancement method that combines Kullback-Leibler (KL) divergence-based non-negative matrix factorization (NMF) and a hidden Markov model (NMF-HMM). With the integration of the HMM, the temporal dynamics information of speech signals can be taken into account. This method includes a training stage and an enhancement stage. In the training stage, the sum of the Poisson distribution, leading to the KL divergence measure, is used as the observation model for each state of the HMM. This ensures that a computationally efficient multiplicative update can be used for the parameter update of this model. In the online enhancement stage, a novel minimum mean square error estimator is proposed for the NMF-HMM. This estimator can be implemented using parallel computing, reducing the time complexity. Moreover, compared to the traditional NMF-based speech enhancement methods, the experimental results show that our proposed algorithm improved the short-time objective intelligibility and perceptual evaluation of speech quality by 5% and 0.18, respectively.

Keywords: Speech enhancement, Non-negative matrix factorization, Hidden Markov model, Minimum mean-square error, Kullback-Leibler divergence

1 Introduction

Single-channel speech enhancement technology is being widely used in our daily lives, such as in speech coding, teleconferencing, hearing aids, mobile communication, and automated robust speech recognition (ASR) [1, 2]. In general, the purpose of speech enhancement is to remove background noise from an audio source while preserving clean speech. It aims to improve the quality and intelligibility of noisy speech [3]. Currently, single-channel speech enhancement is an active topic of research.

During the past decades, many different monaural speech enhancement approaches have been proposed [2, 4]. In an environment with additive noise, the simplest approach to speech enhancement is the spectral subtraction algorithm [5], which subtracts the estimated noise spectrum from the observed signal to acquire the desired clean speech. Other unsupervised methods, such as the signal subspace algorithm [6–9], Wiener filtering [10], minimum mean square error (MMSE) spectral amplitude estimator [11], and log-MMSE spectral amplitude estimator [12], are effective strategies for speech enhancement when the noise is stationary. These methods have low computational complexity and have been widely applied in various areas. However, these approaches cannot always achieve satisfactory performance for non-stationary noise and usually introduce musical noise because they do not make the best use of the prior information of the speech and noise [13]. Moreover, most unsupervised methods are based on the statistical properties of the speech and noise signals. However, it is difficult to meet these properties in actual noisy scenarios [14].

*Correspondence: yaxi@create.aau.dk

[2] Capturi A/S, Søren Frichs Vej 44D, 8230 Aarhus, Denmark
Full list of author information is available at the end of the article

Therefore, supervised speech enhancement approaches have been developed. For instance, Kavalekalam et al. [15] proposed a codebook-based Kalman filter speech enhancement method, which performs a listening test and shows significant improvement for speech intelligibility. In addition, Srinivasan et al. [16] proposed a codebook-driven speech enhancement algorithm for non-stationary noise. In this work, the auto-regressive (AR) spectrum shape codebooks of speech and noise were pre-trained. In the enhancement stage, the codebooks could be used to build a Wiener filter to conduct speech enhancement. Inspired by this research, many other codebook-based speech enhancement approaches have been developed [17, 18]. Furthermore, an auto-regressive hidden Markov model (ARHMM) [19, 20] has also been shown to be an effective supervised speech enhancement method because it considers the temporal information of the speech signal.

In recent years, advances in deep learning techniques [21, 22], specifically, deep neural networks (DNNs), have significantly promoted the development of speech enhancement [23]. These methods usually rely on fewer assumptions [3, 14, 23] between the noise and clean speech, so they have huge potential to achieve better speech enhancement performance. Xu et al. [3, 14] applied a feedforward multilayer perceptron (MLP) to map log-power spectrum (LPS) features of clean speech given noisy LPS input; the enhanced speech could be obtained directly by waveform reconstruction. Compared to the MMSE estimator [12], this method achieved better performance in various noisy environments. Wang et al. [24, 25] also utilized an MLP to estimate the ideal ratio mask (IRM) and ideal binary mask (IBM) in conducting speech enhancement and also achieved satisfactory performance. Motivated by this work, researchers has used different DNN structures to conduct speech enhancement, such as a fully convolutional neural network (FCN) [26], deep recurrent neural networks (DRNN) [27, 28], and generative adversarial networks (GANs) [29, 30]. These methods could help ASR systems achieve higher recognition accuracy in noisy environments. However, generalization is always a problem that needs to be considered for these DNN-based algorithms [31, 32].

A non-negative matrix factorization (NMF)-based speech enhancement algorithm [33–35] can also be viewed as a kind of supervised speech enhancement method. NMF-based methods usually include a training and enhancement stage. In [36], a mask-based NMF speech enhancement method was proposed. In the training stage, the basis matrix of clean speech and noise was trained. In the enhancement stage, the activation matrix could be acquired by combining the trained basis matrix and noisy signal. The mask was then estimated to conduct the speech enhancement. Additionally, an NMF-based denoising scheme was described in [37, 38], which added a heuristic term to the cost function, so the NMF coefficients could be adjusted according to the long-term levels of the signals. A parametric NMF method for speech enhancement was proposed in [17]. This method applied the AR coefficient and codebook to build the basis matrix. This strategy effectively improved the speech intelligibility. Moreover, some DNN-based NMF methods represent an effective strategy for conducting speech enhancement [39, 40]. In general, the basis matrix could be acquired using the traditional NMF method, and the activation matrix could be estimated by applying a DNN, which improved the accuracy of the estimated activation matrix. Thus, it could achieve a higher perceptual evaluation of speech quality (PESQ) [41] and short-time objective intelligibility (STOI) [42] scores than traditional NMF-based speech enhancement methods. The combination of DNN and NMF could also help the ASR system achieve a lower word error rate (WER) in noisy environments. In [43], a DNN-NMF-based method achieved excellent performance in the Computational Hearing in Multisource Environments (CHiME)-3 challenge. To capture temporal information, some HMM-based NMF speech enhancement methods have been proposed. Mohammadiha et al. [44] proposed a supervised and unsupervised NMF speech enhancement method. In [44], an HMM was used for modeling the temporal change of different noise types. In [45], a non-negative factorial HMM was used to model sound mixtures and showed superior performance in source separation tasks. In [46], an HMM-DNN NMF speech enhancement algorithm was proposed, which applied a clustering method to acquire the HMM-based basis matrix and used the Viterbi algorithm to obtain the ideal state label for the DNN training. In the enhancement stage, the DNN was used to find the corresponding state to conduct speech enhancement.

In this paper, we propose a novel NMF-HMM speech enhancement method based on the Kullback-Leibler (KL) divergence, expanding on our preliminary work [47]. Our preliminary work has briefly verified the effectiveness of an NMF-HMM for speech enhancement [47, 48], but the effect of the parameters for the model was not considered. This is very important to optimize the algorithm performance. Additionally, its performance in various noisy environments was also not investigated. In this paper, we expand our preliminary research on these two aspects. Compared to other HMM-based methods [44, 45, 49], our method uses the HMM to capture the temporal dynamics of the speech and noise signal. Moreover, we use the sum of the Poisson distribution as the state-conditioned likelihood for the HMM, rather than

the general Gaussian mixture model (GMM), because the sum of the Poisson distribution leads to the KL divergence measure. KL divergence is a mainstream measure in NMF, and its parameter update rule is identical to the multiplicative update rule. This ensures that the parameter update is computationally efficient during the training stage. In the enhancement stage, in contrast with previous works [44, 45], we propose a novel NMF-HMM-based MMSE estimator to perform the online enhancement. A major benefit of the proposed algorithm is that the activation matrix could be updated by parallel computing in the online stage. This could effectively reduce computational time. In this paper, we also show a more detailed algorithm derivation towards the preliminary NMF-HMM-based algorithm [47]. Moreover, the proposed method was compared with other state-of-the-art speech enhancement algorithms, which further indicated the advantages of the proposed algorithm.

The rest of this paper is organized as follows. First, we will briefly review the general NMF-based speech enhancement method with KL divergence in Section 2. The proposed HMM-based signal model will be introduced in Section 3, and the more detailed offline parameter learning will be explained in Section 4. The details of the proposed MMSE estimator and online speech enhancement process will be given in Section 5. The experimental comparison and analysis of results will be illustrated in Section 5, and we will draw conclusions in Section 6.

2 NMF-based speech enhancement method with KL divergence

In this section, we will briefly review the NMF-based speech enhancement with KL divergence. Under the additive noise assumption, the noisy signal model can be expressed as:

$$y(t) = s(t) + m(t), \tag{1}$$

where $y(t)$, $s(t)$, and $m(t)$ denote the noisy signal, clean speech, and noise, respectively, and t is the time index. With (1), the short-time Fourier transform (STFT) of $y(t)$ can be written as:

$$Y(f,n) = S(f,n) + M(f,n), \tag{2}$$

where $Y(f, n)$, $S(f,n)$, and $M(f,n)$ denote the frequency spectra of $y(t)$, $s(t)$, and $m(t)$, respectively. Here, $f \in [1,F]$ and $n \in [1,N]$ denote the frequency bin and time frame indices, respectively. Collecting the F frequency bins and N time frames, we define the magnitude spectrum matrices $\mathbf{Y}_N$, $\mathbf{S}_N$, and $\mathbf{M}_N$, where $\mathbf{Y}_N = [\mathbf{y}_1, \cdots, \mathbf{y}_n, \cdots, \mathbf{y}_N]$ and $\mathbf{y}_n = [|Y(1,n)|, \cdots, |Y(f,n)|, \cdots, |Y(F,n)|]^T$ and also $\mathbf{s}_n$ and $\mathbf{m}_n$ are defined similarly to $\mathbf{y}_n$. Additionally, $\mathbf{S}_N$ and $\mathbf{M}_N$ are defined similarly to $\mathbf{Y}_N$; we assume that $\mathbf{Y}_N = \mathbf{S}_N + \mathbf{M}_N$. The classical NMF-based speech enhancement has two stages: training and enhancement. In the training stage, the clean speech basis matrix $\overline{\mathbf{W}}$ and noise basis matrix $\ddot{\mathbf{W}}$ are trained using clean speech and noise databases, respectively. Many cost functions have been proposed for NMF, such as KL divergence [34], Itakura-Saito (IS) divergence [50], β divergence, and Euclidean distance [51]. In this paper, we focus on using the KL divergence measure. There are two reasons for this choice. First, compared with other cost functions, the best speech enhancement performance can be achieved using the KL divergence-based NMF with the magnitude spectrum [52]. Second, the efficient multiplicative update (MU) rule of the KL divergence-based NMF can be also derived statistically using the expectation maximization (EM) algorithm [53]. For the two matrices $\mathbf{B}$ and $\hat{\mathbf{B}}$, the KL divergence measure is defined as:

$$\mathrm{KL}(\mathbf{B}|\hat{\mathbf{B}}) = \sum_{i,j}(b_{i,j}\log(b_{i,j}/\hat{b}_{i,j}) - b_{i,j} + \hat{b}_{i,j}), \tag{3}$$

where $b_{i,j}$ and $\hat{b}_{i,j}$ denote the elements from the ith row and jth column of the matrices $\mathbf{B}$ and $\hat{\mathbf{B}}$, respectively. Using speech basis matrix training as an example, the cost function of the KL divergence-based NMF for training $\overline{\mathbf{W}}$ can be written as:

$$(\overline{\mathbf{W}}, \overline{\mathbf{H}}) = \arg\min_{\overline{\mathbf{W}},\overline{\mathbf{H}}} \mathrm{KL}(\mathbf{S}_N|\overline{\mathbf{W}} \times \overline{\mathbf{H}}). \tag{4}$$

Noise basis matrix training is similar to speech basis matrix training. In [34], it is derived that $\overline{\mathbf{W}}$ and $\overline{\mathbf{H}}$ can be obtained iteratively using the following multiplicative update rules:

$$\overline{\mathbf{W}} \leftarrow \overline{\mathbf{W}} \odot \frac{\frac{\mathbf{S}_N}{\overline{\mathbf{W}}\times\overline{\mathbf{H}}}\overline{\mathbf{H}}^T}{\mathbf{1}\overline{\mathbf{H}}^T}, \tag{5}$$

$$\overline{\mathbf{H}} \leftarrow \overline{\mathbf{H}} \odot \frac{\overline{\mathbf{W}}^T\frac{\mathbf{S}_N}{\overline{\mathbf{W}}\times\overline{\mathbf{H}}}}{\overline{\mathbf{W}}^T\mathbf{1}}, \tag{6}$$

where $\odot$ and all divisions are element-wise multiplication and division operations, respectively, and $\mathbf{1}$ is a matrix of ones with the same size as $\mathbf{S}_N$. In the enhancement stage, the noisy speech basis matrix $\mathbf{W}$ can be constructed by concatenating the speech and noise basis matrices, $\mathbf{W} = [\overline{\mathbf{W}}, \ddot{\mathbf{W}}]$. The activation matrix $\mathbf{H}$ of the noisy speech can be estimated iteratively by replacing $\mathbf{S}_N$, $\overline{\mathbf{W}}$, and $\overline{\mathbf{H}}$ in (6) with $\mathbf{Y}_N$, $\mathbf{W}$, and $\mathbf{H}$, respectively. The enhanced signal can be obtained using various algorithms [36, 37, 44, 45]. One popular approach is to

use the following Wiener filter-like spectral gain $\mathbf{g}_n^{\text{NMF}}$ function:

$$\mathbf{g}_n^{\text{NMF}} = \frac{\overline{\mathbf{W}}\,\overline{\mathbf{h}}_n}{\overline{\mathbf{W}}\,\overline{\mathbf{h}}_n + \ddot{\mathbf{W}}\,\ddot{\mathbf{h}}_n}, \tag{7}$$

$$\begin{aligned}\mathbf{h}_n &= \left[\overline{\mathbf{h}}_n^T, \ddot{\mathbf{h}}_n^T\right]^T \\ &= \arg\min_{\mathbf{h}_n} \text{KL}(\mathbf{y}_n|\mathbf{W}\mathbf{h}_n),\end{aligned} \tag{8}$$

where (8) can be solved iteratively using (6). Apart from the gradient descent derivation of the MU update rules (5) and (6) presented in [34], it is further shown in [53] that the MU update rules can be derived from a statistical perspective. More specifically, the KL divergence-based NMF can be motivated from the following hierarchical statistical model:

$$\mathbf{S}_N = \sum_{k=1}^{K} \mathbf{C}(k), \tag{9}$$

$$c_{f,n}(k) \sim \mathcal{PO}(c_{f,n}(k); \overline{W}_{f,k}\overline{H}_{k,n}), \tag{10}$$

where $\mathcal{PO}(x;\lambda) = \frac{\lambda^x e^{-\lambda}}{\Gamma(x+1)}$ is the Poisson distribution, $\Gamma(x+1) = x!$ denotes the gamma function for positive integer x, K denotes the number of basis vectors, $\mathbf{C}(k)$ is the latent matrix, and $c_{f,n}(k)$ denotes the element of $\mathbf{C}(k)$ in the fth row and nth column. Note that $c_{f,n}(k)$ is assumed to have a Poisson distribution, which can only be used for discrete variables. However, in practice, this hierarchical statistical model is not limited to discrete variables because the gamma function for continuous variables can be used to replace the factorial calculation [53]. It has been shown in [53] that the iterative update of the parameters $\overline{\mathbf{H}}$ and $\overline{\mathbf{W}}$ using the EM algorithm is identical to the multiplicative update rules shown in (5) and (6).

One of the advantages of the classical NMF-based method for speech enhancement is that the computational efficient MU rules can be applied. However, the temporal dynamical aspects of speech and noise are not taken into account. To incorporate the temporal dynamical information of audio signals, the HMM model is used in [45] for source separation. However, the parameter update rules are computationally complex. Moreover, this method [45] can only perform the offline enhancement. In this paper, we propose an NMF-based speech enhancement algorithm using the HMM to take the temporal aspects of both the speech and noise into account. The proposed approach can achieve efficient parameter updates. Moreover, an online MMSE estimator for speech enhancement is derived. Although other methods also considered the temporal dynamical information for speech enhancement, such as simply stacking multiple frames to a vector [14, 54], using the DRNN [28], and non-negative matrix deconvolution [55], the high computational complexity and the large model size lead to a high storage complexity. In this paper, the proposed method can achieve a higher PESQ score than the referenced DNN-based method for unseen noise and also has a lower complexity than it.

3 HMM-based signal models with the KL divergence

In this section, we present the details of the proposed signal models, including the speech and noise signal models and the noisy signal model.

3.1 Speech and noise signal models

In this work, the same signal model is used for both the clean speech and noise signals, so we will derive the equations using only the clean speech signal. Additionally, we use the overbar ($\bar{\cdot}$) and double dots ($\ddot{\cdot}$) to represent the clean speech and noise, respectively. To consider the temporal dynamic information of the speech and noise, we use the HMM. Following the conditional independence property of the standard HMM [56], the likelihood function can be expressed as follows:

$$p(\mathbf{S}_N;\mathbf{\Phi}) = \sum_{\overline{\mathbf{x}}_N}\prod_{n=1}^{N} p(\mathbf{s}_n|\overline{x}_n)p(\overline{x}_n|\overline{x}_{n-1}), \tag{11}$$

where $\overline{\mathbf{x}}_N = [\overline{x}_1,\cdots,\overline{x}_n,\cdots,\overline{x}_N]^T$ is a collection of states, $\overline{x}_n \in \{1,2,\cdots,\bar{J}\}$ denote the state at the nth frame, and $\bar{J}$ denotes the total number of states. The function $p(\overline{x}_n|\overline{x}_{n-1})$ denotes the state transition probability from state $\overline{x}_{n-1}$ to $\overline{x}_n$ with $p(\overline{x}_1|\overline{x}_0)$ being the initial state probability. $p(\mathbf{S}_n|\overline{x}_n)$ is the state-conditioned likelihood function, and $\mathbf{\Phi}$ is a collection of modeling parameters. Next, we describe the state transition probability and the state-conditioned likelihood function, respectively, for the proposed signal model.

The state transition probability $p(\overline{x}_n|\overline{x}_{n-1})$: Following the standard HMM, we use a first-order Markov chain to model the state transition, that is:

$$p(\overline{x}_n|\overline{x}_{n-1}) = \prod_{i=1}^{\bar{J}}\prod_{j=1}^{\bar{J}} \overline{A}_{i,j}^{\,l(\overline{x}_n=j,\overline{x}_{n-1}=i)}, \tag{12}$$

$$p(\overline{x}_1|\overline{x}_0) = p(\overline{x}_1) = \prod_{j=1}^{\bar{J}} \overline{\pi}_j^{\,l(\overline{x}_1=j)}, \tag{13}$$

where $l(\cdot)$ denotes an indicator function, which is one when the logic expression in the parentheses is true and zero otherwise. In addition, $\overline{A}_{i,j}$ and $\overline{\pi}_j$ denote the

transition probability from state i to state j and the initial probability for the first frame's state $\bar{x}_1$ being state j, respectively. Collecting all the initial and transition probabilities, we can write them into matrix forms, $\overline{\boldsymbol{\pi}} = [\overline{\pi}_1, \cdots, \overline{\pi}_j, \cdots, \overline{\pi}_{\bar{J}}]^T$ and $\overline{\mathbf{A}}$ with $\overline{A}_{i,j}$ being the element at the ith row and jth column. Therefore, the modeling parameters of the HMM can be expressed as $\boldsymbol{\Phi}_{\text{hmm}} = \{\overline{\mathbf{A}}, \overline{\boldsymbol{\pi}}, \bar{J}\}$. The modeling parameters $\overline{\mathbf{A}}$ and $\overline{\boldsymbol{\pi}}$ with a predefined $\bar{J}$ can be trained through the EM algorithm shown in the next section. In the experiments, we investigate the impact of the total number of states $\bar{J}$.

The state-conditioned likelihood function: Next, we present the proposed state-conditioned likelihood function. Motived by the good speech enhancement performance, the computationally efficient MU rule, and the equivalence between the gradient descent derivation and the EM algorithm for the KL divergence-based NMF, we propose to use the statistical model in (9) and (10) to build the state-conditioned likelihood function, that is:

$$\mathbf{s}_n = \sum_{k=1}^{\overline{K}} \overline{\mathbf{c}}_n(k), \tag{14}$$

$$p(\overline{\mathbf{c}}_n(k)|\bar{x}_n) = \prod_{f=1}^{F} \mathcal{PO}(\overline{c}_{f,n}(k); \overline{W}_{f,k}^{\bar{x}_n} \overline{H}_{k,n}^{\bar{x}_n}), \tag{15}$$

where $\overline{K}$ is the number of basis vectors, $\overline{\mathbf{c}}_n(k)$ contains the hidden variables, and $\overline{W}_{k,n}^{\bar{x}_n}$ and $\overline{H}_{k,n}^{\bar{x}_n}$ correspond to the elements of the basis and activation matrices, respectively. By writing $\overline{\mathbf{c}}_n = [\overline{\mathbf{c}}_n(1)^T, \overline{\mathbf{c}}_n(2)^T, \cdots, \overline{\mathbf{c}}_n(\overline{K})^T]^T$ and integrating $\overline{\mathbf{c}}_n$, the state conditioned likelihood function can be written as:

$$\begin{aligned} p(\mathbf{s}_n|\bar{x}_n) &= \int p(\mathbf{s}_n|\overline{\mathbf{c}}_n) p(\overline{\mathbf{c}}_n|\bar{x}_n)\, d\overline{\mathbf{c}}_n \\ &= \prod_{f=1}^{F} \mathcal{PO}(|S(f,n)|; \sum_{k=1}^{\overline{K}} \overline{W}_{f,k}^{\bar{x}_n} \overline{H}_{k,n}^{\bar{x}_n}), \end{aligned} \tag{16}$$

where we use the superposition property of the Poisson random variable [53]. Collecting the unknown parameters $\{\overline{W}_{f,k}^{\bar{x}_n}\}$ and $\{\overline{H}_{k,n}^{\bar{x}_n}\}$, we can write them into matrix forms, $\{\overline{\mathbf{W}}^j\}$ and $\{\overline{\mathbf{H}}^j\}$. Therefore, unlike the traditional NMF using only one basis matrix, the proposed model has $\bar{J}$ basis matrices to be trained. Each basis matrix is intended to capture a specific feature (e.g., a phoneme) of the speech signal. The modeling parameters of the proposed state-conditioned likelihood function can be expressed as $\boldsymbol{\Phi}_{\text{like}} = \{\{\overline{\mathbf{W}}^j\}, \{\overline{\mathbf{H}}^j\}, \overline{K}, \bar{J}\}$. The modeling parameters $\{\overline{\mathbf{W}}^j\}$ and $\{\overline{\mathbf{H}}^j\}$ with predefined $\bar{J}$ and $\overline{K}$ can be trained through the EM algorithm shown in the next section. In the experiments, we investigate the impact of the number of basis vectors $\overline{K}$ and $\bar{J}$. It will also be shown that a multiplicative update rule can be derived for the basis and activation matrices update of the proposed state-conditioned likelihood function.

To summarize, five types of parameters in the parameter set $\boldsymbol{\Phi}$=$\boldsymbol{\Phi}_{\text{hmm}} \cup \boldsymbol{\Phi}_{\text{like}}$ can be identified. They are the transition matrix $\overline{\mathbf{A}}$, initial state probabilities in $\overline{\boldsymbol{\pi}}$, basis matrices of different states $\{\overline{\mathbf{W}}^j\}$, activation matrices of different states $\{\overline{\mathbf{H}}^j\}$, and modeling parameters $\overline{K}$ and $\bar{J}$. In this paper, the modeling parameters $\overline{K}$ and $\bar{J}$ are predefined, the activation matrices $\{\overline{\mathbf{H}}^j\}$ are estimated by online speech enhancement, and the other three types of parameters are obtained using offline learning.

3.2 Noisy speech model

Based on the proposed clean speech and noise signal models (1) and (2), the noisy speech model can be defined. We assume that there are a total of $\ddot{J}$ hidden states for the noise, and the hidden state of the noise is $\ddot{x}_n (\ddot{x}_n \in \{1, 2, \cdots, \ddot{J}\})$. The notations $\ddot{\boldsymbol{\pi}}$ and $\ddot{\mathbf{A}}$ correspond to the initial state probability and transition probability matrix of the noise. Thus, there are a total of $\bar{J} \times \ddot{J}$ hidden states for the noisy speech. Each composite state consists of a pair of states of clean speech $\bar{x}_n$ and noise $\ddot{x}_n$. Thus, if we list the state space for a noisy signal, we have $(\bar{x}_n = 1, \ddot{x}_n = 1), (\bar{x}_n = 1, \ddot{x}_n = 2), \cdots, (\bar{x}_n = 1, \ddot{x}_n = \ddot{J})$;$(\bar{x}_n = 2, \ddot{x}_n = 1)$, $(\bar{x}_n = 2, \ddot{x}_n = 2), \cdots, (\bar{x}_n = 2, \ddot{x}_n = \ddot{J})$; $\cdots$;$(\bar{x}_n = \bar{J}, \ddot{x}_n = 1), (\bar{x}_n = \bar{J}, \ddot{x}_n = 2)$, $\cdots, (\bar{x}_n = \bar{J}, \ddot{x}_n = \ddot{J})$. Moreover, the initial state and transition probability matrices of the noisy speech can be expressed as $\overline{\boldsymbol{\pi}} \otimes \ddot{\boldsymbol{\pi}}$ and $\overline{\mathbf{A}} \otimes \ddot{\mathbf{A}}$, where $\otimes$ denotes the Kronecker product. Finally, the state conditioned likelihood function of the noisy speech can be written as follows:

$$\begin{aligned} &p(\mathbf{y}_n|\bar{x}_n, \ddot{x}_n) = \\ &\prod_{f=1}^{F} \mathcal{PO}(|(Y(f,n)|; \sum_{k=1}^{\overline{K}} \overline{W}_{f,k}^{\bar{x}_n} \overline{H}_{k,n}^{\bar{x}_n} + \sum_{k=1}^{\ddot{K}} \ddot{W}_{f,k}^{\ddot{x}_n} \ddot{H}_{k,n}^{\ddot{x}_n}), \end{aligned} \tag{17}$$

where $\ddot{K}$, $\{\ddot{W}_{f,k}^{\ddot{x}_n}\}$, and $\{\ddot{H}_{f,k}^{\ddot{x}_n}\}$ represent the number of basis vectors, elements of the basis matrices, and the activation matrices for the noise, respectively. We can write $\{\ddot{W}_{f,k}^{\ddot{x}_n}\}$ and $\{\ddot{H}_{k,n}^{\ddot{x}_n}\}$ into matrix forms as $\{\ddot{\mathbf{W}}^j\}$ and $\{\ddot{\mathbf{H}}^j\}$, respectively. Note that we also used the superposition property of Poisson random variables to obtain (17).

4 Methods

4.1 Offline NMF-HMM-based parameter learning

In the offline training stage, the objective is to find the parameter set $\boldsymbol{\Phi}$ that maximizes the likelihood function (11). In general, the EM algorithm [56] can be used to address this problem. Because we use the same model

for the speech and noise, here, we use the clean speech as an example to illustrate the offline parameter learning process. First, we define the complete data set $(\mathbf{S}_N, \bar{\mathbf{x}}_N, \overline{\mathbf{C}}_N)$, where $\overline{\mathbf{C}}_N = [\bar{\mathbf{c}}_1, \bar{\mathbf{c}}_2, \cdots, \bar{\mathbf{c}}_N]$. Thus, using the conditional independence property, the complete data likelihood function can be written as:

$$p(\mathbf{S}_N, \bar{\mathbf{x}}_N, \overline{\mathbf{C}}_N) = \prod_{n=1}^{N} p(\mathbf{s}_n|\bar{\mathbf{c}}_n)p(\bar{\mathbf{c}}_n|\bar{x}_n)p(\bar{x}_n|\bar{x}_{n-1}). \tag{18}$$

Next, we show how the parameter set can be obtained iteratively using the EM algorithm. Moreover, we propose an acceleration strategy to lower the computational and memory complexities. The traditional MU update algorithm for the KL divergence-based NMF can be seen as a special case of the proposed algorithm.

Expectation step: We first calculate the posterior state probability and the joint posterior probability, which can be written as:

$$q(\bar{x}_n) = p(\bar{x}_n|\mathbf{S}_N; \boldsymbol{\Phi}^{i-1}), \tag{19}$$

$$q(\bar{x}_n, \bar{x}_{n-1}) = p(\bar{x}_n, \bar{x}_{n-1}|\mathbf{S}_N; \boldsymbol{\Phi}^{i-1}), \tag{20}$$

where i is the iteration number. The calculation of (19) and (20) can be performed using the forward-backward algorithm [56]. Apart from this, we also need to evaluate the posterior expectation $\mathbb{E}_{\bar{\mathbf{c}}_n|\mathbf{S}_N,\bar{x}_n;\boldsymbol{\Phi}^{i-1}}(\bar{\mathbf{c}}_n)$, which will be used in the maximization step. By using the Bayes rule and the conditional independence property of the proposed model, we have:

$$q(\bar{\mathbf{c}}_n|\bar{x}_n) = p(\bar{\mathbf{c}}_n|\mathbf{S}_N, \bar{x}_n; \boldsymbol{\Phi}^{i-1}) = \frac{p(\mathbf{s}_n|\bar{\mathbf{c}}_n)p(\bar{\mathbf{c}}_n|\bar{x}_n)}{p(\mathbf{S}_N, \bar{x}_n)}. \tag{21}$$

Combining (14) and (15) and following the derivation in [53], we have:

$$\begin{aligned} &q(\bar{\mathbf{c}}_n|\bar{x}_n) = \\ &\prod_{f=1}^{F} \mathcal{M}(\bar{c}_{f,n}(1), \cdots, \bar{c}_{f,n}(\overline{K}); |S(f,n)|, \\ &p_{f,n}^{\bar{x}_n}(1), \cdots, p_{f,n}^{\bar{x}_n}(\overline{K})), \end{aligned} \tag{22}$$

where $\mathcal{M}(\cdot)$ denotes the multinomial distribution and

$$p_{f,n}^{\bar{x}_n}(k) = \frac{\overline{W}_{f,k}^{\bar{x}_n}\overline{H}_{k,n}^{\bar{x}_n}}{\sum_{l=1}^{\overline{K}}\overline{W}_{f,l}^{\bar{x}_n}\overline{H}_{l,n}^{\bar{x}_n}}. \tag{23}$$

Using the properties of the multinomial distribution, the mean can be written as:

$$\mathbb{E}(\bar{c}_{f,n}(k)|\mathbf{S}_N, \bar{x}_n) = |S(f,n)|\frac{\overline{W}_{f,k}^{\bar{x}_n}\overline{H}_{k,n}^{\bar{x}_n}}{\sum_{l=1}^{K}\overline{W}_{f,l}^{\bar{x}_n}\overline{H}_{l,n}^{\bar{x}_n}}. \tag{24}$$

Maximization step: In this step, our objective is to find parameters to maximize the expectation of the logarithm of the complete data likelihood, that is,

$$\boldsymbol{\Phi}^i = \arg\max_{\boldsymbol{\Phi}} \mathbb{E}_{\bar{\mathbf{x}}_N, \overline{\mathbf{C}}_N|\mathbf{S}_N;\boldsymbol{\Phi}^{i-1}}[\log p(\mathbf{S}_N, \bar{\mathbf{x}}_N, \overline{\mathbf{C}}_N)]. \tag{25}$$

The estimators for $\overline{\mathbf{A}}$ and $\overline{\boldsymbol{\pi}}$ are the same as the traditional HMM [56]. For completeness, the results are shown below:

$$\overline{\pi}_j = \frac{q(\bar{x}_1 = j)}{\sum_{o=1}^{\bar{J}} q(\bar{x}_1 = o)}, \tag{26}$$

$$\overline{A}_{o,j} = \frac{\sum_{n=2}^{\overline{N}} q(\bar{x}_n = j, \bar{x}_{n-1} = o)}{\sum_{j=1}^{\bar{J}}\sum_{n=2}^{\overline{N}} q(\bar{x}_n = j, \bar{x}_{n-1} = o)}, \tag{27}$$

where $1 \le o, j \le \bar{J}$. The estimated basis and activation matrices can be derived by setting the derivatives of (25) to zeros, and we can obtain:

$$W_{f,k}^j = \frac{\sum_{n=1}^{N} q(\bar{x}_n = j)\mathbb{E}(\bar{c}_{f,n}(k)|\mathbf{S}_N, \bar{x}_n = j)}{\sum_{n=1}^{N} q(\bar{x}_n = j)H_{k,n}^j}, \tag{28}$$

$$H_{k,n}^j = \frac{\sum_{f=1}^{F} \mathbb{E}(\bar{c}_{f,n}(k)|\mathbf{S}_N, \bar{x}_n = j)}{\sum_{f=1}^{F} W_{f,k}^j}. \tag{29}$$

Acceleration strategy: Although we can directly use the above EM algorithm to update the parameter set, saving the conditional expectation of $\bar{c}_{f,n}(k)$ in (24) requires a great deal of memory. Like [53], we substitute (24) into (28) and (29) and can obtain:

$$W_{f,k}^j \leftarrow \frac{\sum_{n=1}^{N} q(\bar{x}_n = j)\frac{|S(f,n)|\overline{H}_{k,n}^j}{\sum_{l=1}^{\overline{K}}\overline{W}_{f,l}^j\overline{H}_{l,n}^j}}{\sum_{n=1}^{N} q(\bar{x}_n = j)H_{k,n}^j}, \tag{30}$$

$$H_{k,n}^j \leftarrow \frac{\sum_{f=1}^{F} \frac{\overline{W}_{f,k}^j|S(f,n)|}{\sum_{l=1}^{K}\overline{W}_{f,l}^j\overline{H}_{l,n}^j}}{\sum_{f=1}^{F} H_{k,n}^j}. \tag{31}$$

We can further write (30) and (31) in matrix forms:

$$\overline{\mathbf{W}}^{j} \leftarrow \overline{\mathbf{W}}^{j} \odot \frac{\frac{\mathbf{S}_N}{\overline{\mathbf{W}}^{j}\overline{\mathbf{H}}^{j}}\mathbf{\Lambda}(j)(\overline{\mathbf{H}}^{j})^{T}}{\mathbf{1}\mathbf{\Lambda}(j)(\overline{\mathbf{H}}^{j})^{T}}, \tag{32}$$

$$\overline{\mathbf{H}}^{j} \leftarrow \overline{\mathbf{H}}^{j} \odot \frac{(\overline{\mathbf{W}}^{j})^{T}\frac{\mathbf{S}_N}{\overline{\mathbf{W}}^{j}\overline{\mathbf{H}}^{j}}}{(\overline{\mathbf{W}}^{j})^{T}\mathbf{1}}, \tag{33}$$

where $\mathbf{\Lambda}(j) = \mathrm{diag}(q(\bar{x}_1 = j), q(\bar{x}_2 = j), \cdots, q(\bar{x}_N = j))$. By using the proposed acceleration strategy, the computing and saving of the conditional expectation of $\bar{c}_{f,n}(k)$ in (24) is not required. Moreover, the multiplicative update rules for the basis and activation matrices can be obtained, leading to fast computing. In other words, there are more than one basis and active matrices to be estimated in the proposed algorithm. Using acceleration strategy, the different basis and active matrices can be simultaneously estimated. We do not need to estimate them one by one. This reduces the time complexity. Comparing the update rules of the proposed method (32), (33) with the traditional NMF-based method (5), (6), the difference is that the basis vectors update rule (32) for the proposed method takes the posterior state information $\mathbf{\Lambda}(j)$ into account. In fact, if the number of the state is set to one (i.e., $\bar{J} = 1$), the proposed training method is identical to the traditional KL divergence-based NMF approach. Thus, the traditional NMF can be seen as a special case of the proposed algorithm. The entire flow of the offline parameter learning is shown in Algorithm 1. Note that, for stability reasons, each column of $\overline{\mathbf{W}}^{j}$ is normalized to have a unit norm during training.

Algorithm 1: Offline NMF-HMM-based parameter learning

1: Randomly initiate $\overline{\mathbf{W}}^{j}$ and $\overline{\mathbf{H}}^{j}, \bar{j} \in \{1, 2, \cdots, \bar{J}\}$
2: **for** $i = 1, 2, 3, \cdots, I$ **do**
Expectation step:
3: Calculate $p(\mathbf{s}_n|\bar{x}_n)$, $1 \le n \le N$ based on (16)
4: Obtain (19) and (20) using the forward-backward algorithm [56]
Maximization step:
5: Re-estimate $\overline{\boldsymbol{\pi}}$ and $\overline{\mathbf{A}}$ based on (26) and (27)
6: Re-estimate $\overline{\mathbf{W}}^{j}$ and $\overline{\mathbf{H}}^{j}$ based on (32) and (33)
7: **end for**

4.2 Online speech enhancement using the MMSE estimator

4.2.1 MMSE estimator for the NMF-HMM

In this section, we provide a detailed derivation for the proposed MMSE-based online speech enhancement algorithm in the proposed NMF-HMM model. Our objective is to obtain the MMSE estimate of the desired clean speech signal from noisy observation:

$$\hat{\mathbf{s}}_n = \mathbb{E}_{\mathbf{s}_n|\mathbf{Y}_n}(\mathbf{s}_n) = \int \mathbf{s}_n p(\mathbf{s}_n|\mathbf{Y}_n)\, d\mathbf{s}_n. \tag{34}$$

In (34), the posterior probability $p(\mathbf{s}_n|\mathbf{Y}_n)$ can be derived as:

$$\begin{aligned} p(\mathbf{s}_n|\mathbf{Y}_n) &= \frac{p(\mathbf{s}_n, \mathbf{y}_n|\mathbf{Y}_{n-1})}{p(\mathbf{y}_n|\mathbf{Y}_{n-1})} \\ &= \frac{\sum_{\bar{x}_n, \ddot{x}_n} p(\mathbf{s}_n, \mathbf{y}_n|\bar{x}_n, \ddot{x}_n) p(\bar{x}_n, \ddot{x}_n|\mathbf{Y}_{n-1})}{p(\mathbf{y}_n|\mathbf{Y}_{n-1})}, \end{aligned} \tag{35}$$

where we use the conditional independence property of the HMM. The term $p(\bar{x}_n, \ddot{x}_n|\mathbf{Y}_{n-1})$ in (35) can be expressed as:

$$\begin{aligned} &p(\bar{x}_n, \ddot{x}_n|\mathbf{Y}_{n-1}) \\ &= \sum_{\bar{x}_{n-1}, \ddot{x}_{n-1}} p(\bar{x}_n, \ddot{x}_n|\bar{x}_{n-1}, \ddot{x}_{n-1}) p(\bar{x}_{n-1}, \ddot{x}_{n-1}|\mathbf{Y}_{n-1}), \end{aligned} \tag{36}$$

where the first term after the summation is the state transition probability for a noisy signal, and the second term is the forward probability that can be acquired using the well-known forward algorithm [56]. By applying the Bayes rule, the term $p(\mathbf{s}_n, \mathbf{y}_n|\bar{x}_n, \ddot{x}_n)$ in (35) can be further written as:

$$p(\mathbf{s}_n, \mathbf{y}_n|\bar{x}_n, \ddot{x}_n) = p(\mathbf{s}_n|\mathbf{y}_n, \bar{x}_n, \ddot{x}_n) p(\mathbf{y}_n|\bar{x}_n, \ddot{x}_n). \tag{37}$$

Substituting (37) for (35), the posterior probability can be re-written as:

$$p(\mathbf{s}_n|\mathbf{Y}_n) = \sum_{\bar{x}_{n-1}, \ddot{x}_{n-1}} \omega_{\bar{x}_n, \ddot{x}_n} p(\mathbf{s}_n|\mathbf{y}_n, \bar{x}_n, \ddot{x}_n), \tag{38}$$

where the weight $0 \le \omega_{\bar{x}_n, \ddot{x}_n} \le 1$ is defined as:

$$\omega_{\bar{x}_n, \ddot{x}_n} = \frac{p(\mathbf{y}_n|\bar{x}_n, \ddot{x}_n) p(\bar{x}_n, \ddot{x}_n|\mathbf{Y}_{n-1})}{\sum_{\bar{x}_n, \ddot{x}_n} p(\mathbf{y}_n|\bar{x}_n, \ddot{x}_n) p(\bar{x}_n, \ddot{x}_n|\mathbf{Y}_{n-1})}. \tag{39}$$

Thus, by combining (34) and (38), the proposed HMM-based MMSE estimator can be expressed as:

$$\hat{\mathbf{s}}_n = \sum_{\bar{x}_n, \ddot{x}_n} \omega_{\bar{x}_n, \ddot{x}_n} \int \mathbf{s}_n p(\mathbf{s}_n|\mathbf{y}_n, \bar{x}_n, \ddot{x}_n)\, d\mathbf{s}_n. \tag{40}$$

Instead of obtaining the posterior probability density function (PDF) $p(\mathbf{s}_n|\mathbf{y}_n, \bar{x}_n, \ddot{x}_n)$ directly, we derive the formula for the joint posterior PDF of the clean speech and noise first, that is:

$$
\begin{aligned}
&p(\mathbf{s}_n, \mathbf{m}_n|\mathbf{y}_n, \bar{x}_n, \ddot{x}_n) \\
&= \frac{p(\mathbf{y}_n|\mathbf{s}_n, \mathbf{m}_n)p(\mathbf{s}_n, \mathbf{m}_n|\bar{x}_n, \ddot{x}_n)}{p(\mathbf{y}_n|\bar{x}_n, \ddot{x}_n)} \\
&= \frac{p(\mathbf{y}_n|\mathbf{s}_n, \mathbf{m}_n)p(\mathbf{s}_n|\bar{x}_n)p(\mathbf{m}_n|\ddot{x}_n)}{p(\mathbf{y}_n|\bar{x}_n, \ddot{x}_n)}.
\end{aligned} \tag{41}
$$

By using (1), we can express the likelihood function $p(\mathbf{y}_n|\mathbf{s}_n, \mathbf{m}_n)$ as $p(\mathbf{y}_n|\mathbf{s}_n, \mathbf{m}_n) = \delta(\mathbf{y}_n - \mathbf{s}_n - \mathbf{m}_n)$, where $\delta(\cdot)$ denotes the Dirac delta function, which is defined by $\delta(0) = +\infty$, and $\delta(x) = 0$ when $x \neq 0$. Furthermore, $\int_{-\infty}^{+\infty} \delta(x)\, dx = 1$. The prior probability $p(\mathbf{s}_n|\bar{x}_n)$ and $p(\mathbf{m}_n|\ddot{x}_n)$ can be estimated by using (16). Following the derivation in [53], we can verify that the joint posterior PDF can be expressed in terms of the multinomial distribution as:

$$
\begin{aligned}
&p(\mathbf{s}_n, \mathbf{m}_n|\mathbf{y}_n, \bar{x}_n, \ddot{x}_n) = \\
&\prod_{f=1}^{F} \mathcal{M}(|S(f,n)|, |M(f,n)|; \\
&|Y(f,n)|, p_{f,n}(\bar{x}_n, \ddot{x}_n), q_{f,n}(\bar{x}_n, \ddot{x}_n)),
\end{aligned} \tag{42}
$$

where $p_{f,n}(\bar{x}_n, \ddot{x}_n)$ and $q_{f,n}(\bar{x}_n, \ddot{x}_n)$ are defined as:

$$
\begin{aligned}
&p_{f,n}(\bar{x}_n, \ddot{x}_n) = \\
&\frac{\sum_{k=1}^{\bar{K}} \bar{W}_{f,k}^{\bar{x}_n} \bar{H}_{k,n}^{\bar{x}_n}}{\sum_{k=1}^{\bar{K}} \bar{W}_{f,k}^{\bar{x}_n} \bar{H}_{k,n}^{\bar{x}_n} + \sum_{k=1}^{\ddot{K}} \ddot{W}_{f,k}^{\ddot{x}_n} \ddot{H}_{k,n}^{\ddot{x}_n}},
\end{aligned} \tag{43}
$$

where $q_{f,n}(\bar{x}_n, \ddot{x}_n) = 1 - p_{f,n}(\bar{x}_n, \ddot{x}_n)$. Therefore, the integral term in (40) can be expressed as:

$$
\begin{aligned}
&\int \mathbf{s}_n p(\mathbf{s}_n|\mathbf{y}_n, \bar{x}_n, \ddot{x}_n)\, d\mathbf{s}_n \\
&= \int \mathbf{s}_n \int p(\mathbf{s}_n, \mathbf{m}_n|\mathbf{y}_n, \bar{x}_n, \ddot{x}_n)\, d\mathbf{m}_n\, d\mathbf{s}_n \\
&= \mathbf{y}_n \odot \mathbf{p}_n(\bar{x}_n, \ddot{x}_n),
\end{aligned} \tag{44}
$$

where $\mathbf{p}_n(\bar{x}_n, \ddot{x}_n) = [p_{1,n}(\bar{x}_n, \ddot{x}_n), \cdots, p_{F,n}(\bar{x}_n, \ddot{x}_n)]^T$, and we used the marginal mean property of the multinomial distribution. Combining (40) and (44), the MMSE estimator can be expressed as:

$$
\hat{\mathbf{s}}_n = \mathbf{y}_n \odot \mathbf{g}_n, \tag{45}
$$

$$
\mathbf{g}_n = \sum_{\bar{x}_n, \ddot{x}_n} \omega_{\bar{x}_n, \ddot{x}_n} \mathbf{p}_n(\bar{x}_n, \ddot{x}_n), \tag{46}
$$

where $\mathbf{g}_n$ can be viewed as the spectral gain vector for the proposed model. Comparing the proposed gain vector $\mathbf{g}_n$ with the traditional NMF-based gain vector [36], we find that the proposed gain vector is a weighted sum of each state's gain, which is in the Wiener filtering form as the traditional NMF gain (7).

4.2.2 Online estimation of activation matrices

After obtaining the trained basis matrices $\bar{W}_{f,k}^{\bar{x}_n}$ and $\ddot{W}_{f,k}^{\ddot{x}_n}$ for both the clean speech and noise in the training stage, we need to obtain the online estimates of the activation parameters $\bar{H}_{f,k}^{\bar{x}_n}$ and $\ddot{H}_{f,k}^{\ddot{x}_n}$ to acquire the gain in (45) and (46). The activation matrices are estimated by maximizing the logarithm of the state-conditioned likelihood function (17), which is equivalent to:

$$
\mathbf{h}_n(\bar{x}_n, \ddot{x}_n) = \arg\min_{\mathbf{h}_n} \ \mathrm{KL}(\mathbf{y}_n|[\bar{\mathbf{W}}^{\bar{x}_n}, \ddot{\mathbf{W}}^{\ddot{x}_n}]\mathbf{h}_n), \tag{47}
$$

$$
\mathbf{h}_n(\bar{x}_n, \ddot{x}_n) = [\bar{\mathbf{h}}_n(\bar{x}_n, \ddot{x}_n)^T, \ddot{\mathbf{h}}_n(\bar{x}_n, \ddot{x}_n)^T]^T, \tag{48}
$$

where the clean and noise activation matrices for the state $(\bar{x}_n, \ddot{x}_n)$ are defined as $\bar{\mathbf{h}}_n(\bar{x}_n, \ddot{x}_n) = [\bar{H}_{1,n}^{\bar{x}_n}, \bar{H}_{2,n}^{\bar{x}_n}, \cdots, \bar{H}_{\bar{K},n}^{\bar{x}_n}]^T$, and $\ddot{\mathbf{h}}_n(\bar{x}_n, \ddot{x}_n) = [\ddot{H}_{1,n}^{\ddot{x}_n}, \ddot{H}_{2,n}^{\ddot{x}_n}, \cdots, \ddot{H}_{\ddot{K},n}^{\ddot{x}_n}]^T$. The activation matrix (48) can be obtained iteratively by using the multiplicative update rule in Eq. (6). Note that parallel computing can be used to reduce the time complexity when obtaining the activation matrices for different states. It can be readily shown that when $\bar{J} = \ddot{J} = 1$, the gain vectors for the proposed algorithm (46) and the standard NMF (7) are identical, that is, $\mathbf{g}_n = \mathbf{g}_n^{\mathrm{NMF}}$. The entire flow of the proposed MMSE-based online speech enhancement algorithm is illustrated by Algorithm 2.

Algorithm 2: MMSE-based online speech enhancement

1: Input magnitude spectrum: $\mathbf{Y}_n$
2: Initiate $\bar{\boldsymbol{\pi}} \otimes \ddot{\boldsymbol{\pi}}$ and $\bar{\mathbf{A}} \otimes \ddot{\mathbf{A}}$
3: **for** $n = 1, 2, 3, \cdots, N$ **do**
4: Initiate $\mathbf{h}_n(\bar{x}_n, \ddot{x}_n)$
5: Based on (6) and (48), obtain the iterative estimation $\mathbf{h}_n(\bar{x}_n, \ddot{x}_n)$
6: Calculate $p(\mathbf{y}_n|\bar{x}_n, \ddot{x}_n)$ based on (17)
7: Apply the forward algorithm and combine (36) and (39) to acquire $\omega_{\bar{x}_n, \ddot{x}_n}$
7: Obtain $\mathbf{p}_n(\bar{x}_n, \ddot{x}_n)$ using (43)
8: Calculate the spectral gain $\mathbf{g}_n$ using (46)
9: By equation(45), estimate the clean speech $\hat{\mathbf{s}}_n$
10: **end for**

5 Experimental results and discussion

In this section, we report on the investigation and evaluation of the proposed algorithm using various experiments. First, we investigated the effect of different parameter settings for the proposed model, that is, the number of states and basis vectors of clean speech and noise, respectively. Second, we compared the proposed NMF-HMM with other state-of-the-art speech enhancement methods to demonstrate the effectiveness of the proposed algorithm. In this work, the PESQ score [41], ranging from − 0.5 to 4.5, was used to quantify

the enhanced speech quality. The version of the PESQ model used was the International Telecommunication Union (ITU) standard P.862 [57]. The implementation code was provided by [2]. The STOI score [42], ranging from 0 to 1, was used to measure speech intelligibility.

5.1 Experimental data preparation

In this study, the proposed algorithm was evaluated using the Texas Instruments/Massachusetts Institute of Technology (TIMIT) database [58], 100 environmental noises [59], office noise[1], and the NoiseX-92 database [60]. During the training stage, all 4620 utterances from the TIMIT training database were used to train the proposed NMF-HMM model for clean speech. For the experiments in Section 5.2, the Babble, F16, Factory, and White noises from the NoiseX-92 database were used to train the NMF-HMM model. For the experiments in Section 5.2, 200 utterances from the TIMIT test set, including 1680 utterances, were randomly chosen to build the test database. Four types of noise were then added at four different SNR levels (− 5, 0, 5, and 10 dB). The noise types of the testing set were the same as the training set, but there was no overlap between the signals in the two sets. In total, $200 \times 4 \times 4 = 3200$ utterances were used for the evaluation. For the experiments in Section 5.3, we conducted extensive experiments; the Babble and F16 noises from the NoiseX-92 database and 90 environmental noises (N1–N90 in [59]) were used to train the NMF-HMM model for the noise dictionary. In the test stage, 200 utterances from the TIMIT test set, including 1680 utterances, were randomly chosen to build three test databases. The first test database included 10 unseen environmental noises from [59] (N91–N100). The second included unseen office noise, and the third test database was built from 25 seen environmental noises in [59] (N18–N43). In all three test databases, the noise was added at four different SNR levels (− 5, 0, 5, and 10 dB). All the algorithms were evaluated using the same test dataset. In all experiments, the sound signals were down-sampled to 16 kHz. The frame length was set to 1024 samples (64 ms) with a frame shift of 512 samples (32 ms). The size of STFT was 1024 points with a Hanning window. Furthermore, the maximum number of iterations was set to 30 in the training stage and 15 in the online speech enhancement stage for the proposed NMF-HMM algorithm.

5.2 Analyses of the number of states and basis vectors

As explained in Sections 3 and 4, four parameters are needed to be pre-defined in our proposed NMF-HMM-based speech enhancement algorithm. These parameters were the number of states ($\bar{J}$ and $\ddot{J}$) and basis vectors ($\overline{K}$ and $\ddot{K}$) for the clean speech and noise. In this section, we report on the investigation of the effects of these parameters in our proposed method and the choice of suitable parameters for the later experiments.

5.2.1 HMM states analysis

First, before the state analysis, we want to indicate that using temporal dynamics can effectively help NMF obtain a better SE performance. To verify this point, we use the traditional NMF-based speech enhancement (T-NMF) [36] as the reference method. T-NMF is a special case of NMF-HMM when $\bar{J} = 1$ and $\ddot{J} = 1$. T-NMF does not include the temporal dynamics information. The transition matrix A is a non-informational matrix in T-NMF. For a fair comparison, we keep that the total numbers of clean speech basis vectors ($\overline{K} \times \bar{J}$) for the NMF-HMM and T-NMF method [36] are the same. For the T-NMF, the number of clean speech basis vectors $\overline{K}$ is varied as 25, 125, 250, 500, and 1000. For the NMF-HMM, the $\overline{K}$ is fixed to 25 and $\bar{J}$ is varied as 1, 5, 10, 20, and 40. The number of noise basis vectors for both the proposed NMF-HMM and T-NMF is fixed to 70, and the number of noise states for the NMF-HMM is fixed to 1. In this experiment, we use the average STOI and PESQ scores of 3200 utterances as the performance metrics. The experimental results are shown in Fig. 1. As can be seen, the T-NMF can achieve the best performance when $\overline{K} = 25$. However, its performance degraded with the increasing of number of basis vectors due to overfitting. By contrast, NMF-HMM achieves higher PESQ and STOI scores with an increasing number of the clean speech basis vectors by taking the temporal dynamics into account using the HMM model, which indicates that temporal dynamics can improve the NMF's SE performance.

5.2.2 States and basis vector analysis for clean speech

Next, we investigated the effect of the number of clean speech states $\bar{J}$ and basis vector $\overline{K}$ to the proposed model. The number of noise states was set to 1 (i.e., $\ddot{J} = 1$) for the proposed NMF-HMM. The number of basis vectors for the noise was fixed to $\ddot{K} = 70$, respectively. The number of clean speech states was chosen as 1, 5, 10, 20, and 40. Additionally, the number of clean speech basis vector was chosen as 5, 10, 25, and 50. The enhancement performance was evaluated by the PESQ and STOI scores.

Tables 1 and 2 show the average STOI and PESQ score in different SNRs. It can be seen that if the number of basis vectors $\overline{K}$ is fixed, there is a higher PESQ and STOI score with the increasing of clean state $\bar{J}$. This indicated the benefits of using the temporal dynamics in NMF

[1] https://www.youtube.com/watch?v=D7ZZp8XuUTE

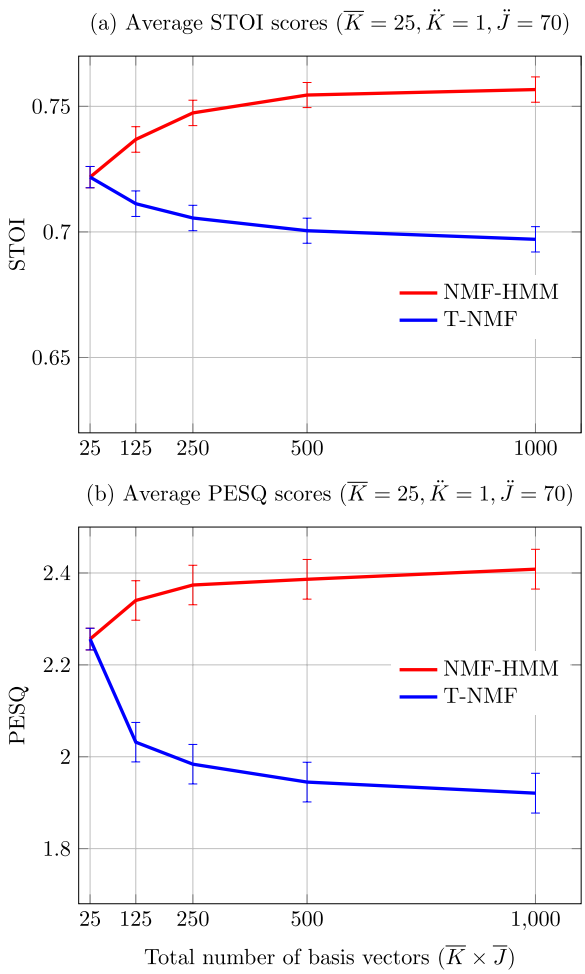

Fig. 1 Performance of the NMF-HMM and T-NMF using different numbers of clean speech basis vectors

we choose $\overline{J} = 40$ and $\overline{K} = 25$ to perform the following experiments.

5.2.3 States and basis vector analysis for noise

In this part, we evaluated the effect of noise states $\ddot{J}$ and basis vector $\ddot{K}$ to the proposed model. Here, the number of clean states and basis vectors was set to 40 and 25 ($\overline{J} = 40$, $\ddot{K} = 25$), respectively, which is based on the previous experimental results. The number of noise states was chosen as 1, 2, 5, and 10. In addition, the number of noise basis vector was chosen as 10, 20, 40, and 70.

Tables 3 and 4 show the experimental results for the average STOI and PESQ score in different SNRs. We can find that the PESQ and STOI have an increasing trend with the increasing of noise state $\ddot{J}$ when the number of noise basis vectors $\ddot{K}$ is fixed. Moreover, if the $\ddot{J}$ is fixed, $\ddot{K} = 70$ can achieve the highest PESQ score but the STOI score is slightly lower than $\ddot{K} = 40$. Based on the experimental results, we select $\overline{J} = 40, \ddot{J} = 10, \overline{K} = 25, \ddot{K} = 40$ for the rest of the experiments because the model have the less parameters when $\ddot{K} = 40$. Furthermore, there is a higher STOI when $\ddot{K} = 40$ and the PESQ difference is not obvious between the $\ddot{K} = 40$ and $\ddot{K} = 70$.

5.3 Overall evaluation

In this section, we report on the comparison of the proposed NMF-HMM speech enhancement method with state-of-the-art speech enhancement methods. We chose the optimally modified log-spectral amplitude (OM-LSA) method [61] with improved minima controlled recursive averaging (IMCRA) noise estimator [62]; variable span linear filters method [7] (SLF-NMF),

Table 1 Average STOI scores (%) comparisons of different clean speech states and basis vectors ($\ddot{J} = 1, \ddot{K} = 70$)

Parameters	$\overline{K} = 5$	$\overline{K} = 10$	$\overline{K} = 25$	$\overline{K} = 50$
Noisy	**69.14** (± 0.51)			
NMF-HMM, $\overline{J} = 1$(T-NMF)	65.00 (± 0.43)	69.29 (± 0.44)	72.71 (± 0.48)	73.32 (± 0.49)
NMF-HMM, $\overline{J} = 5$	68.66 (± 0.42)	71.93 (± 0.45)	73.94 (± 0.47)	74.02 (± 0.49)
NMF-HMM, $\overline{J} = 10$	69.71 (± 0.42)	72.74 (± 0.45)	74.39 (± 0.47)	74.37 (± 0.50)
NMF-HMM, $\overline{J} = 20$	71.14 (± 0.43)	73.51 (± 0.45)	74.76 (± 0.48)	74.87 (± 0.50)
NMF-HMM, $\overline{J} = 40$	71.81 (± 0.44)	73.66 (± 0.45)	75.00 (± 0.48)	74.73 (± 0.51)

model. Additionally, if the clean state $\overline{J}$ is fixed, we can find that HMM can achieve the best speech enhancement performance when $\overline{K} = 25$. A higher $\overline{K}$ can lead to a worse speech enhancement performance due to overfitting. Therefore, based on these experimental results, which uses the parametric NMF [17] for estimating the statistics; temporal-NMF [49]; convolutive NMF (CNMF) [55, 63]; DNN [64]; and log-MMSE [65] algorithm as the reference methods. For the SLF-NMF, the maximum SNR filter was applied, and the number of

Table 2 Average PESQ scores (%) comparisons of different clean speech states and basis vectors ($\ddot{J} = 1, \ddot{K} = 70$)

Parameters	$\overline{K} = 5$	$\overline{K} = 10$	$\overline{K} = 25$	$\overline{K} = 50$
Noisy	**2.02** (± 0.03)			
NMF-HMM, $\overline{J} = 1$(T-NMF)	2.12 (± 0.03)	2.18 (± 0.03)	2.21 (± 0.02)	2.18 (± 0.02)
NMF-HMM, $\overline{J} = 5$	2.27 (± 0.03)	2.31 (± 0.03)	2.32 (± 0.02)	2.29 (± 0.02)
NMF-HMM, $\overline{J} = 10$	2.31 (± 0.03)	2.35 (± 0.03)	2.35 (± 0.03)	2.30 (± 0.02)
NMF-HMM, $\overline{J} = 20$	2.36 (± 0.03)	2.39 (± 0.02)	2.36 (± 0.02)	2.32 (± 0.02)
NMF-HMM, $\overline{J} = 40$	2.38 (± 0.02)	2.41 (± 0.02)	2.39 (± 0.02)	2.33 (± 0.02)

Table 3 Average STOI scores (%) comparisons of different noise states and basis vectors ($\overline{J} = 40, \overline{K} = 25$)

Parameters	$\ddot{K} = 10$	$\ddot{K} = 20$	$\ddot{K} = 40$	$\ddot{K} = 70$
Noisy	**69.14** (± 0.51)			
NMF-HMM, $\ddot{J} = 1$	74.51 (± 0.51)	74.71 (± 0.51)	75.03 (± 0.49)	75.00 (± 0.48)
NMF-HMM, $\ddot{J} = 2$	75.00 (± 0.51)	75.30 (± 0.50)	75.51 (± 0.49)	75.33 (± 0.47)
NMF-HMM, $\ddot{J} = 5$	75.44 (± 0.51)	75.77 (± 0.50)	76.05 (± 0.47)	75.15 (± 0.46)
NMF-HMM, $\ddot{J} = 10$	75.56 (± 0.50)	76.11 (± 0.49)	76.27 (± 0.48)	75.70 (± 0.46)

Table 4 Average PESE scores (%) comparisons of different noise states and basis vectors ($\overline{J} = 40, \overline{K} = 25$)

Parameters	$\ddot{K} = 10$	$\ddot{K} = 20$	$\ddot{K} = 40$	$\ddot{K} = 70$
Noisy	**2.02** (± 0.03)			
T-NMF, $\ddot{J} = 1$	2.28 (± 0.03)	2.31 (± 0.03)	2.36 (± 0.02)	2.39 (± 0.02)
NMF-HMM, $\ddot{J} = 2$	2.29 (± 0.03)	2.33 (± 0.04)	2.37 (± 0.04)	2.40 (± 0.03)
NMF-HMM, $\ddot{J} = 5$	2.31 (± 0.03)	2.34 (± 0.04)	2.39 (± 0.03)	2.40 (± 0.03)
NMF-HMM, $\ddot{J} = 10$	2.32 (± 0.03)	2.36 (± 0.03)	2.40 (± 0.02)	2.41 (± 0.02)

eigenvectors was set to one. The variable span linear filters reference code can be found in [7]. The codebook size of clean speech and noise was set to 64 and 8, respectively. The other SLF-NMF parameter settings were the same as NMF-HMM. For the temporal-NMF, all the parameter settings were the same as the work in [49], which ensured that the temporal-NMF could achieve the best speech enhancement performance. For the CNMF, the related settings were similar to the CNMF in [40]. For the DNN, we used the DNS baseline [64] as the reference method, which is one of the state of the art speech enhancement algorithm. The OM-LSA and log-MMSE were state-of-the-art unsupervised speech enhancement methods. while the SLF-NMF and temporal-NMF were state-of-the-art NMF-based speech enhancement methods. The temporal-NMF also considered the temporal information like our methods.

The performance of the NMF-HMM, DNN, temporal-NMF, CNMF, SLF-NMF, log-MMSE, and OM-LSA were evaluated using the test set. Figure 2 shows the average PESQ scores with 95% confidence intervals of these algorithms for 25 types of seen noise. As can be seen, the SLF-NMF had the worst performance among these algorithms. Temporal-NMF and CNMF achieved a higher score than SLF-NMF, which indicated the benefits of temporal information for speech enhancement. Moreover, except for DNS baseline, the proposed NMF-HMM outperformed other enhancement algorithms in all the SNR scenarios. Furthermore, in low SNR scenarios (e.g., − 5–5 dB), the average PESQ score improvement of the proposed NMF-HMM was larger than 0.5 against the other algorithms.

Figures 3 and 4 show the PESQ result under an unseen noise environment, which indicates that NMF-HMM could always achieve a higher PESQ score than the reference methods at all four SNRs except for DNS baseline.

The results of the STOI scores with 95% confidence intervals for various algorithms are provided in Table 5. As can be seen, the temporal-NMF, CNMF, and NMF-HMM had higher STOI scores than SLF-NMF under three different test datasets, which illustrates the benefits of considering speech temporal information. In general, NMF-HMM achieved the highest STOI score, better than the referenced NMF-based methods (temporal-NMF, CNMF, and SLF-NMF) for seen and unseen noise. In addition, the DNS baseline achieved a better STOI score than NMF-HMM.

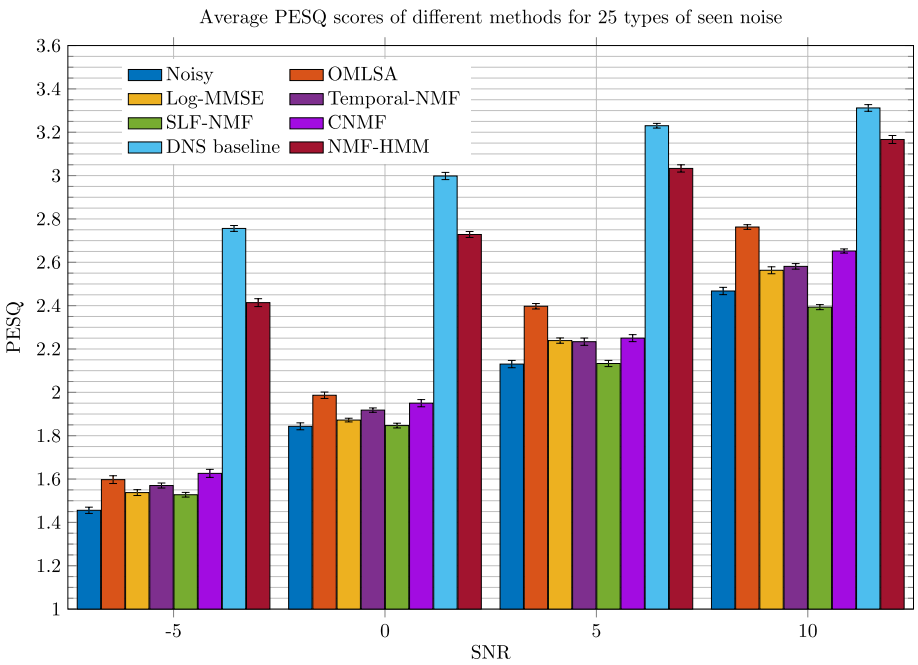

Fig. 2 Average PESQ scores of different methods for 25 types of seen noise

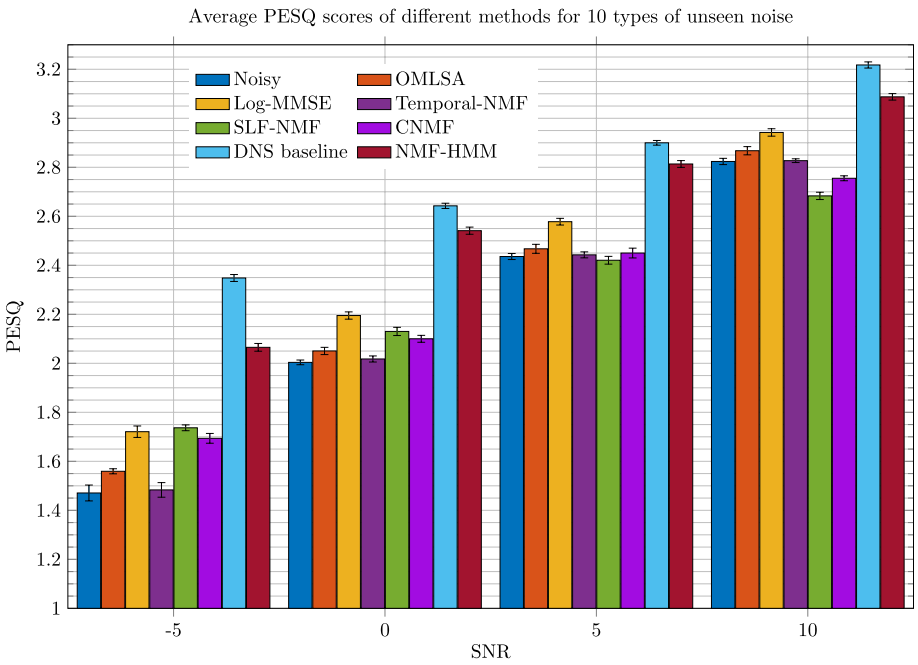

Fig. 3 Average PESQ scores of different methods for 10 types of unseen noise

In general, for these non-DNN-based speech enhancement algorithm, the proposed method can achieve the best speech enhancement performance. Moreover, DNS baseline can achieve the highest speech enhancement score. In the future work, we think that a DNN-based strategy can be combine with proposed algorithm to improve to accuracy of basis vector estimation. As a result, our algorithm can achieve a better speech enhancement performance.

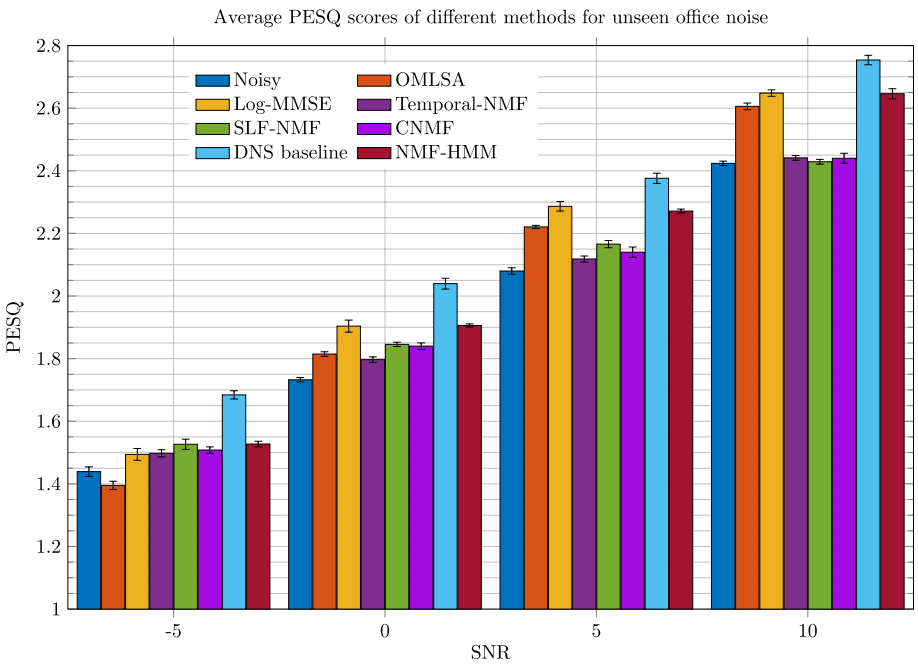

Fig. 4 Average PESQ scores of different methods for unseen office noise

Table 5 Comparison of STOI scores (%) for various algorithms under different SNRs using different types of noise

Test type	Method	− 5	0	5	10
Unseen 10 types of noise	**Noisy**	76.97 (± 1.45)	84.24 (± 0.96)	90.07 (± 0.68)	94.16 (± 0.49)
	Log-MMSE	75.86 (± 1.54)	83.67 (± 1.01)	89.72 (± 0.70)	93.85 (± 0.48)
	OMLSA	75.88 (± 1.52)	83.58 (± 1.01)	89.51 (± 0.72)	93.62 (± 0.55)
	Temporal-NMF	77.21 (± 1.45)	84.39 (± 0.96)	90.15 (± 0.68)	94.19 (± 0.49)
	SLF-NMF	69.35 (± 1.78)	77.01 (± 1.28)	82.11 (± 1.09)	85.72 (± 0.94)
	CNMF	77.12 (± 1.51)	83.02 (± 1.13)	86.01 (± 1.02)	89.44 (± 0.91)
	NMF-HMM	78.58 (± 1.34)	84.76 (± 0.84)	88.39 (± 0.58)	90.88 (± 0.43)
	DNS baseline	**81.84** (± 1.36)	**86.91** (± 1.09)	**91.44** (± 0.75)	**94.67** (± 0.55)
Unseen office noise	**Noisy**	49.91 (± 1.33)	61.03 (± 1.40)	72.80 (± 1.27)	82.57 (± 1.05)
	Log-MMSE	46.46 (± 1.50)	58.75 (± 1.57)	71.09 (± 1.40)	81.31 (± 1.15)
	OMLSA	44.97 (± 1.52)	58.14 (± 1.63)	71.52 (± 1.44)	82.29 (± 1.14)
	Temporal-NMF	49.70 (± 1.46)	61.79 (± 1.47)	73.48 (± 1.29)	83.05 (± 1.05)
	SLF-NMF	48.92 (± 1.58)	60.84 (± 1.54)	70.95 (± 1.35)	79.21 (± 1.12)
	CNMF	48.43 (± 1.47)	60.97 (± 1.46)	71.45 (± 1.12)	80.03 (± 0.97)
	NMF-HMM	50.06 (± 1.72)	63.02 (± 1.61)	74.56 (± 1.32)	82.55 (± 0.88)
	DNS baseline	**54.22** (± 1.49)	**66.46** (± 1.01)	**77.58** (± 0.89)	**86.18** (± 0.50)
textbfSeen 25 types of noise	**Noisy**	73.65 (± 0.82)	81.36 (± 1.03)	87.64 (± 0.84)	92.48 (± 0.60)
	Log-MMSE	71.96 (± 1.40)	80.13 (± 1.20)	87.04 (± 0.94)	92.08 (± 0.68)
	OMLSA	73.86 (± 1.38)	81.58 (± 1.18)	87.90 (± 0.91)	92.45 (± 0.66)
	Temporal-NMF	75.76 (± 1.34)	83.22 (± 1.09)	89.03 (± 0.88)	93.46 (± 0.58)
	SLF-NMF	65.76 (± 1.58)	73.49 (± 1.33)	79.06 (± 1.18)	83.14 (± 1.04)
	CNMF	76.23 (± 1.38)	84.12 (± 1.11)	89.55 (± 0.97)	91.06 (± 0.62)
	NMF-HMM	81.49 (± 1.66)	87.02 (± 1.35)	90.28 (± 0.77)	91.84 (± 0.51)
	DNS baseline	**81.95** (± 1.76)	**87.34** (± 1.15)	**91.53** (± 0.75)	**94.77** (± 0.53)

6 Conclusion

In this work, we proposed and analyzed an NMF-HMM-based speech enhancement algorithm that applies the sum of the Poisson distribution, leading to the KL divergence measure, as the observation model for each state of the HMM. The computationally efficient multiplicative update rule is used to conduct parameter updates during the training stage for this proposed method. Moreover, using the HMM, the temporal dynamic information of speech signals can be captured in this method. Furthermore, we detailed the derivation of the proposed NMF-HMM-based MMSE estimator to conduct online speech enhancement. Parallel computation can be applied for the proposed estimator, so we can effectively reduce the time complexity during the online speech enhancement stage. With experiments, a suitable number of state basis vectors for the proposed NMF-HMM were found. Our experimental results also indicated that the proposed algorithm could outperform state-of-the-art NMF-based and unsupervised speech enhancement methods. In the future work, a DNN-based strategy can be considered to improve the accuracy of basis vector estimation. As a result, our algorithm can achieve a better speech enhancement performance.

Acknowledgements
This work was supported by Innovation Fund Denmark (Grant No.9065-00046).

Authors' contributions
All authors participate in methodology discussion, experimental design, and paper writing. The authors read and approved the final manuscript.

Funding
Innovation Fund Denmark (Grant No.9065-00046).

Availability of data and materials
Not applicable.

Declarations

Competing interests
The authors declare that they have no competing interests.

Author details
[1]CREATE, Aalborg University, Rendsburggade 14, 9000 Aalborg, Denmark.
[2]Capturi A/S, Søren Frichs Vej 44D, 8230 Aarhus, Denmark.

Received: 24 January 2022 Accepted: 22 August 2022
Published online: 08 September 2022

References

1. J. Li, L. Deng, Y. Gong, R. Haeb-Umbach, An overview of noise-robust automatic speech recognition. IEEE/ACM Trans. Audio Speech Lang. Process. **22**(4), 745–777 (2014)
2. P.C. Loizou, *Speech Enhancement: Theory and Practice* (CRC Press, Boca Raton, 2013)
3. Y. Xu, J. Du, L.-R. Dai, C.-H. Lee, An experimental study on speech enhancement based on deep neural networks. IEEE Signal Process. Lett. **21**(1), 65–68 (2013)
4. I. Cohen, S. Gannot, in *Springer Handbook of Speech Processing. Spectral enhancement methods* (Springer, Berlin, Heidelberg, 2008) p. 873–902
5. S. Boll, Suppression of acoustic noise in speech using spectral subtraction. IEEE Trans. Acoust. Speech Signal Process. **27**(2), 113–120 (1979)
6. K.B. Christensen, M.G. Christensen, J.B. Boldt, F. Gran, in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. Experimental study of generalized subspace filters for the cocktail party situation* (IEEE, Shanghai, 2016), p. 420–424
7. J.R. Jensen, J. Benesty, M.G. Christensen, Noise reduction with optimal variable span linear filters. IEEE/ACM Trans. Audio Speech Lang. Process. **24**(4), 631–644 (2015)
8. Y. Ephraim, H.L. Van Trees, A signal subspace approach for speech enhancement. IEEE Trans. Speech Audio Process. **3**(4), 251–266 (1995)
9. F. Jabloun, B. Champagne, Incorporating the human hearing properties in the signal subspace approach for speech enhancement. IEEE Trans. Speech Audio Process. **11**(6), 700–708 (2003)
10. J. Lim, A. Oppenheim, All-pole modeling of degraded speech. IEEE Trans. Acoust. Speech Signal Process. **26**(3), 197–210 (1978)
11. Y. Ephraim, D. Malah, Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator. IEEE Trans. Acoust. Speech Signal Process. **32**(6), 1109–1121 (1984)
12. Y. Ephraim, D. Malah, Speech enhancement using a minimum mean-square error log-spectral amplitude estimator. IEEE Trans. Acoust. Speech Signal Process. **33**(2), 443–445 (1985)
13. A. Hussain, M. Chetouani, S. Squartini, A. Bastari, F. Piazza, in *Progress in nonlinear speech processing. An overview, Nonlinear speech enhancement* (Springer, Berlin, Heidelberg, 2007), p. 217–248
14. Y. Xu, J. Du, L.-R. Dai, C.-H. Lee, A regression approach to speech enhancement based on deep neural networks. IEEE/ACM Trans. Audio Speech Lang. Process. **23**(1), 7–19 (2014)
15. M.S. Kavalekalam, J.K. Nielsen, J.B. Boldt, M.G. Christensen, Model-based speech enhancement for intelligibility improvement in binaural hearing aids. IEEE/ACM Trans. Audio Speech Lang. Process. **27**(1), 99–113 (2018)
16. S. Srinivasan, J. Samuelsson, W.B. Kleijn, Codebook-based bayesian speech enhancement for nonstationary environments. IEEE Trans. Audio Speech Lang. Process. **15**(2), 441–452 (2007)
17. M.S. Kavalekalam, J.K. Nielsen, L. Shi, M.G. Christensen, J. Boldt, in *Proc. European Signal Processing Conf. Online parametric NMF for speech enhancement* (IEEE, Rome, 2018), p. 2320–2324
18. Q. He, F. Bao, C. Bao, Multiplicative update of auto-regressive gains for codebook-based speech enhancement. IEEE/ACM Trans. Audio Speech Lang. Process. **25**(3), 457–468 (2016)
19. D.Y. Zhao, W.B. Kleijn, HMM-based gain modeling for enhancement of speech in noise. IEEE Trans. Audio Speech Lang. Process. **15**(3), 882–892 (2007)
20. F. Deng, C. Bao, W.B. Kleijn, Sparse hidden Markov models for speech enhancement in non-stationary noise environments. IEEE/ACM Trans. Audio Speech Lang. Process. **23**(11), 1973–1987 (2015)
21. Y. Bengio et al., Learning deep architectures for AI. Found. Trends® Mach. Learn. **2**(1), 1–127 (2009)
22. G.E. Hinton, S. Osindero, Y.-W. Teh, A fast learning algorithm for deep belief nets. Neural Comput. **18**(7), 1527–1554 (2006)
23. D. Wang, J. Chen, Supervised speech separation based on deep learning: an overview. IEEE/ACM Trans. Audio Speech Lang. Process. **26**(10), 1702–1726 (2018)
24. Y. Wang, A. Narayanan, D. Wang, On training targets for supervised speech separation. IEEE/ACM Trans. Audio Speech Lang. Process. **22**(12), 1849–1858 (2014)
25. A. Narayanan, D. Wang, in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. Ideal ratio mask estimation using deep neural networks for robust speech recognition* (IEEE, Vancouver, 2013), p. 7092–7096
26. S.R. Park, J. Lee, A fully convolutional neural network for speech enhancement. arXiv preprint arXiv:1609.07132. (2016)
27. H. Jacobsson, Rule extraction from recurrent neural networks: Ataxonomy and review. Neural Comput. **17**(6), 1223–1263 (2005)
28. P.-S. Huang, M. Kim, M. Hasegawa-Johnson, P. Smaragdis, Joint optimization of masks and deep recurrent neural networks for monaural source separation. IEEE/ACM Trans. Audio Speech Lang. Process. **23**(12), 2136–2147 (2015)
29. I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, et al., in *Proc. Advances in Neural Inform. Process. Syst. Generative adversarial nets* (Communications of the ACM, US, 2014), p. 2672–2680

30. S. Pascual, A. Bonafonte, J. Serra, Segan: Speech enhancement generative adversarial network. arXiv preprint arXiv:1703.09452. (2017)
31. M. Kolbæk, Z.-H. Tan, J. Jensen, Speech intelligibility potential of general and specialized deep neural network based speech enhancement systems. IEEE/ACM Trans. Audio Speech Lang. Process. **25**(1), 153–167 (2016)
32. Y. Xiang, C. Bao, A parallel-data-free speech enhancement method using multi-objective learning cycle-consistent generative adversarial network. IEEE/ACM Trans. Audio Speech Lang. Process. **28**, 1826–1838 (2020)
33. D.D. Lee, H.S. Seung, Learning the parts of objects by non-negative matrix factorization. Nature. **401**(6755), 788–791 (1999)
34. D.D. Lee, H.S. Seung, in *Proc. Advances in Neural Inform. Process. Syst. Algorithms for non-negative matrix factorization* (Communications of the ACM, US, 2001), p. 556–562
35. K. Shimada, Y. Bando, M. Mimura, K. Itoyama, K. Yoshii, T. Kawahara, Unsupervised speech enhancement based on multichannel nmf-informed beamforming for noise-robust automatic speech recognition. IEEE/ACM Trans. Audio Speech Lang. Process. **27**(5), 960–971 (2019)
36. E.M. Grais, H. Erdogan, in *Int. Conf. Digital Signal Process. Single channel speech music separation using nonnegative matrix factorization and spectral masks* (IEEE, Corfu, 2011), p. 1–6
37. K.W. Wilson, B. Raj, P. Smaragdis, in *Proc Interspeech. Regularized non-negative matrix factorization with temporal dependencies for speech denoising* (ICSA, Brisbane, 2008)
38. S. Nie, S. Liang, H. Li, X. Zhang, Z. Yang, W.J. Liu, L.K. Dong, in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. Exploiting spectro-temporal structures using NMF for DNN-based supervised speech separation* (IEEE, Shanghai, 2016), p. 469–473
39. T.G. Kang, K. Kwon, J.W. Shin, N.S. Kim, NMF-based target source separation using deep neural network. IEEE Signal Process. Lett. **22**(2), 229–233 (2014)
40. S. Nie, S. Liang, W. Liu, X. Zhang, J. Tao, Deep learning based speech separation via nmf-style reconstructions. IEEE/ACM Trans. Audio Speech Lang. Process. **26**(11), 2043–2055 (2018)
41. A.W. Rix, J.G. Beerends, M.P. Hollier, A.P. Hekstra, in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs*, vol. 2 (IEEE, Salt Lake City, 2001), p. 749–752
42. 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)
43. T.T. Vu, B. Bigot, E.S. Chng, in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. Combining non-negative matrix factorization and deep neural networks for speech enhancement and automatic speech recognition* (IEEE, Shanghai, 2016), p. 499–503
44. N. Mohammadiha, P. Smaragdis, A. Leijon, Supervised and unsupervised speech enhancement using nonnegative matrix factorization. IEEE Trans. Audio Speech Lang. Process. **21**(10), 2140–2151 (2013)
45. G.J. Mysore, P. Smaragdis, B. Raj, in *International conference on latent variable analysis and signal separation. Non-negative hidden Markov modeling of audio with application to source separation* (Springer, Malo, 2010), p. 140–148
46. Z. Wang, X. Li, X. Wang, Q. Fu, Y. Yan, in *Proc. Interspeech. A DNN-HMM approach to non-negative matrix factorization based speech enhancement* (ICSA, Pittsburgh, 2016), p. 3763–3767
47. Y. Xiang, L. Shi, J.L. Højvang, M.H. Rasmussen, M.G. Christensen, in *Proc. Interspeech. An NMF-HMM speech enhancement method based on Kullback-Leibler divergence* (ICSA, Shanghai, 2020), p. 2667–2671
48. Y. Xiang, L. Shi, J.L. Højvang, M.H. Rasmussen, M.G. Christensen, in *Proc. IEEE Int. Conf. coust., Speech, Signal Process. A novel NMF-HMM speech enhancement algorithm based on poisson mixture model* (IEEE, Toronto, 2021), p. 721–725
49. C. Févotte, J. Le Roux, J.R. Hershey, in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. Non-negative dynamical system with application to speech and audio* (IEEE, Vancouver, 2013), p. 3158–3162
50. C. Févotte, N. Bertin, J.-L. Durrieu, Nonnegative matrix factorization with the itakura-saito divergence: with application to music analysis. Neural Comput. **21**(3), 793–830 (2009)
51. C. Févotte, J. Idier, Algorithms for nonnegative matrix factorization with the β-divergence. Neural Comput. **23**(9), 2421–2456 (2011)
52. D. FitzGerald, M. Cranitch, E. Coyle, *On the use of the beta divergence for musical source separation* (IET digital library, Dublin, 2009)
53. A.T. Cemgil, Bayesian inference for nonnegative matrix factorisation models. Computational intelligence and neuroscience. **2009**, 1–17 (2009)
54. D. Baby, J.F. Gemmeke, T. Virtanen, et al., in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. Exemplar-based speech enhancement for deep neural network based automatic speech recognition* (IEEE, South Brisbane, 2015), p. 4485–4489
55. P. Smaragdis, Convolutive speech bases and their application to supervised speech separation. IEEE Trans. Audio Speech Lang. Process. **15**(1), 1–12 (2006)
56. L.E. Baum, An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes. Inequalities. **3**(1), 1–8 (1972)
57. I.-T. Recommendation, *Perceptual evaluation of speech quality (PESQ): an objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs. Rec. ITU-T P* (IEEE, US, 2001), p. 862
58. J.S. Garofolo, L.F. Lamel, W.M. Fisher, J.G. Fiscus, D.S. Pallett, DARPA TIMIT acoustic-phonetic continous speech corpus CD-ROM. NIST speech disc 1-1.1. NASA STI/Recon technical report n. **93**, (1993)
59. G. Hu, D. Wang, A tandem algorithm for pitch estimation and voiced speech segregation. IEEE Trans. Audio Speech Lang. Process. **18**(8), 2067–2079 (2010)
60. A. Varga, H.J. Steeneken, Assessment for automatic speech recognition: II. NOISEX-92: a database and an experiment to study the effect of additive noise on speech recognition systems. Speech Commun **12**(3), 247–251 (1993)
61. I. Cohen, B. Berdugo, Speech enhancement for non-stationary noise environments. Signal Process. **81**(11), 2403–2418 (2001)
62. I. Cohen, Noise spectrum estimation in adverse environments: improved minima controlled recursive averaging. IEEE Trans. Speech Audio Process. **11**(5), 466–475 (2003)
63. P.D. O'grady, B.A. Pearlmutter, Discovering speech phones using convolutive non-negative matrix factorisation with a sparseness constraint. Neurocomputing **72**(1–3), 88–101 (2008)
64. S. Braun, I. Tashev, in *International Conference on Speech and Computer. Data augmentation and loss normalization for deep noise suppression* (Springer, Petersburg, 2020), p. 79–86
65. T. Gerkmann, R.C. Hendriks, Unbiased MMSE-based noise power estimation with low complexity and low tracking delay. IEEE Trans. Audio Speech Lang. Process. **20**(4), 1383–1393 (2011)

Publisher's Note