# Dictionary Learning for Sparse Representation of Signals With Hidden Markov Model Dependency

S. AKHAVAN[a,b], F. BAGHESTANI[a], P. KAZEMI[a], A. KARAMI[a], H. SOLTANIAN-ZADEH[a,c]

[a] *School of Electrical and Computer Engineering, University of Tehran, Tehran, Iran*
[b] *Univ. Grenoble Alpes, Grenoble INP, CNRS, GIPSA-lab, Grenoble, France*
[c] *Medical Image Analysis Lab., Henry Ford Health System, Detroit, MI, USA*

**Abstract**

The goal of dictionary learning algorithms is factorizing the matrix of training signals $\mathbf{Y}$ with $K$ signals, into the dictionary matrix $\mathbf{D}$ and the coefficient matrix $\mathbf{X}$ which is a sparse matrix. The common approach among the algorithms is minimizing the representation error subject to the sparseness of $\mathbf{X}$ using alternation minimization method following 1) the sparsification and 2) the dictionary update steps. In this approach, when $\mathbf{D}$ is fixed and $\mathbf{X}$ must be estimated (the sparsification step), the minimization problem is divided to $K$ different sparse recovery problems because the training signals are assumed to be independent. However, in some signals such as medical signals, this assumption is not correct and the signals are not independent. Therefore, using the current strategy does not lead to the correct estimation of the parameters. In this study, we investigate the dictionary learning problem for sparse representation when there is hidden Markov model (HMM) dependency among the training signals. We propose an approach to improve the performance of the dictionary learning algorithms in the mentioned scenario. The proposed approach is not an independent dictionary learning algorithm. It is a general approach that we can employ for existing dictionary learning algorithms to improve their performance in learning from signals with HMM dependency. We confirm the efficiency of the proposed approach using simulations, and also, present a real application in medical signals for the considered scenario.

*Keywords:* dictionary; coefficient matrix; sparse representation; hidden Markov model; transition probability matrix

## 1. Introduction

### *1.1. Preliminary*

In many signal processing applications, it is desired to linearly map the signals into a new domain, and then, process the new representation of the signals. This linear mapping is performed using a matrix called dictionary. In general, the dictionaries can be categorized into two groups; 1) non-adaptive, and 2)

adaptive. In the non-adaptive category, the dictionaries are pre-defined such as Fourier or Discrete Cosine Transform (DCT), while in the adaptive category, the dictionaries are learned from the signals like in Principal Component Analysis (PCA). It has been shown that adaptive dictionaries outperform non-adaptive ones in many applications such as classification [30], compression [26, 39] and denoising [8, 40, 33].

In the adaptive category, learning a dictionary which can sparsely represent the signals has attracted a lot of attention during the last decade. Sparse component analysis [10, 6], compressed sensing [19], image reconstruction [13, 17], registration [2], and compression [14, 41] are from the applications of dictionary learning for sparse representation of the signals. Mathematically, the goal is factorizing the matrix of the signals $\mathbf{Y} \in \mathbb{R}^{M \times K}$ to the dictionary $\mathbf{D} \in \mathbb{R}^{M \times N}$ which is often an overcomplete matrix, and the coefficient matrix $\mathbf{X} \in \mathbb{R}^{N \times K}$ which is sparse, as schematically shown in Fig. 1. It should be noted that the subscript $K$ shows the number of training signals in matrix $\mathbf{Y}$, and the number of rows in overcomplete dictionaries is less than the number of columns ($M << N$) which makes the problem underdetermined.

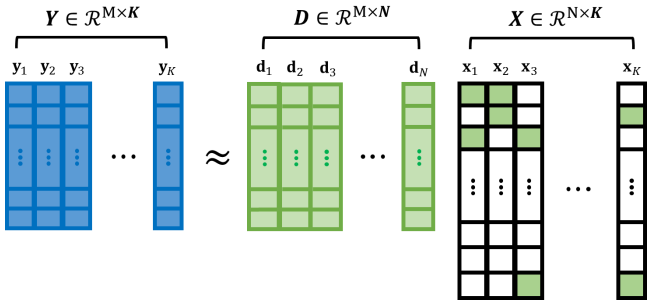


Figure 1: Dictionary learning for sparse representation of the signals.

By considering $\mathbf{Y} = [\mathbf{y}_1\, \mathbf{y}_2 \ldots \mathbf{y}_K]$, $\mathbf{D} = [\mathbf{d}_1\, \mathbf{d}_2 \ldots \mathbf{d}_N]$ and $\mathbf{X} = [\mathbf{x}_1\, \mathbf{x}_2 \ldots \mathbf{x}_K]$, the dictionary learning for sparse representation of the signals can be performed by solving the following constrained optimization problem:

$$\begin{aligned} \{\mathbf{D}, \mathbf{X}\} = & \underset{\mathbf{D},\mathbf{X}}{\operatorname{argmin}} \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2 \\ s.t. \quad & \|\mathbf{x}_k\|_0 \leq N_0, \quad 1 \leq k \leq K \\ & \|\mathbf{d}_n\|_2 = 1, \quad 1 \leq n \leq N \end{aligned} \tag{1}$$

where $\|.\|_F$, $\|.\|_0$, and $\|.\|_2$ respectively show the Frobenius norm, the $l_0$ norm (the number of non-zero entries) and the Euclidean norm. The first constraint forces the columns of the coefficient matrix to be sparse, and $N_0$ denotes the sparsity level (e.g., $N_0 = 2$ in Fig. 1). In fact, we use the sparsity constraint to find a unique solution for the system of equations expressed in (1). Due to the non-convexity of $l_0$ norm, it is often substituted by $l_1$ norm, which is a well-known relaxation for $l_0$ norm. The second constraint omits the scaling ambiguity, and each column of the dictionary is usually called an atom.

Alternation minimization is a common approach to solve the constrained optimization problem stated in (1). This means that some initializations are considered for the parameters. Then, the following two steps, i.e., 1) sparsification and 2) dictionary update steps, are alternately performed until convergence of the parameters.

*1.1.1. Sparsification*

Assuming that the dictionary ($\mathbf{D}$) is fixed, the following constrained optimization problem is solved in this step:

$$\begin{aligned} \mathbf{X} = & \underset{\mathbf{X}}{\operatorname{argmin}} \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2 \\ s.t. \ & \|\mathbf{x}_k\|_0 \leq N_0, \ \ 1 \leq k \leq K \end{aligned} \tag{2}$$

The observed signals are usually assumed to be independent, hence, (2) is decoupled to the following constrained optimization problems for $k = 1, 2, ..., K$:

$$\begin{aligned} \mathbf{x}_k = & \underset{\mathbf{x}_k}{\operatorname{argmin}} \|\mathbf{y}_k - \mathbf{D}\mathbf{x}_k\|_F^2 \\ & s.t. \ \ \|\mathbf{x}_k\|_0 \leq N_0 \end{aligned} \tag{3}$$

Any of the sparse recovery algorithms such as Orthogonal Matching Pursuit (OMP) [27] or Basis Pursuit (BP) [9] can be used to perform the sparsification step.

*1.1.2. Dictionary Update*

Assuming that the coefficient matrix ($\mathbf{X}$) is fixed, the following constrained optimization problem is solved in this step:

$$\begin{aligned} \mathbf{D} = & \underset{\mathbf{D}}{\operatorname{argmin}} \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2 \\ s.t. \ & \|\mathbf{d}_n\|_2 = 1, \ \ 1 \leq n \leq N \end{aligned} \tag{4}$$

The main difference among the existing dictionary learning algorithms is in the dictionary update step. In the following, we briefly explain how some well-known algorithms perform the dictionary update step.

*1.2. State of The Art Algorithms*

Some of the algorithms simultaneously update the atoms of the dictionary, while some others sequentially estimate them. Method of Optimal Directions (MOD)[15] is one of the algorithms that simultaneously estimate the atoms of the dictionary in the dictionary update step, and the following closed form solution is obtained for the dictionary:

$$\mathbf{D} = \mathbf{Y}\mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1} \rightarrow \mathbf{d}_n = \frac{\mathbf{d}_n}{\|\mathbf{d}_n\|_2} \quad 1 \leq n \leq N \tag{5}$$

Unlike MOD, K-Singular Value Decomposition (K-SVD) [3] estimates the atoms of the dictionary sequentially. In K-SVD when an atom is updated,

the non-zero entries of the corresponding row in the coefficient matrix are also updated using SVD. Since only the non-zero entries of the coefficient matrix change in the dictionary update step, the support of the coefficient matrix is kept fixed in this step. If we respectively show the $n^{th}$ row of the coefficient matrix and its non-zero entries by $(\mathbf{x}_{[n]})^T$ and $(\mathbf{x}^r_{[n]})^T$, the following constrained optimization problem is solved for updating $\mathbf{d}_n$ and $\mathbf{x}^r_{[n]}$ in K-SVD:

$$\begin{aligned} \{\mathbf{d}_n, \mathbf{x}^r_{[n]}\} &= \underset{\mathbf{d}_n, \mathbf{x}^r_{[n]}}{\operatorname{argmin}} \|\mathbf{E}^r_n - \mathbf{d}_n(\mathbf{x}^r_{[n]})^T\|_F^2 \\ s.t. \quad \|\mathbf{d}_n\|_2 &= 1, \quad \mathbf{E}_n = \mathbf{Y} - \sum_{i \neq n} \mathbf{d}_i(\mathbf{x}_{[i]})^T \end{aligned} \tag{6}$$

where $\mathbf{E}^r_n$ consists of the columns of $\mathbf{E}_n$ which are corresponding to the non-zero entries of $\mathbf{x}_{[n]}$.

Simultaneous codeword optimization (SimCO) method [11] is a generalized version of K-SVD that improves the performance of the dictionary learning. By keeping the support of the coefficient matrix fixed, a few atoms instead of an atom and the associated rows of the coefficient matrix are simultaneously estimated in SimCO at a time in the dictionary update step.

The idea of MOD and K-SVD have been combined, and Multiple Dictionary Update (MDU) method has been proposed in [35]. In this method, the dictionary and the non-zero entries of the coefficient matrix are updated in the dictionary update step by solving the following constrained optimization problem:

$$\begin{aligned} \{\mathbf{D}, \mathbf{X}\} &= \underset{\mathbf{D}, \mathbf{X}}{\operatorname{argmin}} \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2 \\ s.t. \quad \mathbf{X} \odot \mathbf{M} &= 0, \quad \|\mathbf{d}_n\|_2 = 1, \quad 1 \leq n \leq N \end{aligned} \tag{7}$$

where $\odot$ denotes the entry-wise (Hadamard) multiplication, and $\mathbf{M}$ is a mask matrix of zeros and ones which forces the zero entries of $\mathbf{X}$ to remain intact. This optimization problem can be solved using alternating minimization. Since the objective function and the constraints are convex with respect to each of the parameters, a closed form solution can be easily obtained for the parameters in each iteration. It has been shown that MDU leads to better performance of the dictionary learning.

The focus of this study is on the algorithms which use the data in batch mode for learning the dictionary. It is worth mentioning that some of the algorithms perform the dictionary learning online. This means that the dictionary and the coefficient matrix are updated by coming the new data. Recursive least square dictionary learning [34], online dictionary learning for sparse coding [21], and correlation based online dictionary learning [25] are from the pioneer online algorithms. The main idea in online algorithms is similar to one employed in batch mode algorithms. For instance in [25], the atoms related to the sparse representation of the new data are first updated while the other atoms are kept intact. Then, the sparse representations of the previous data that have common atoms with the new data are updated.

### 1.3. Contribution of This Study

Since there is no prior information about the generation of the signals, the dictionary learning algorithms assume that the training signals are independent, and as mentioned in (2) and (3), the sparsification step is individually performed on each column of $\mathbf{Y}$. In some applications such as medical signals and images, e.g., diffusion weighted images or extracellular depth electroencephalography (EEG) recordings [5, 1], the generation of the signals is according to a specific dynamic model, and hence, the training signals are not independent. Therefore, decoupling (2) to (3) is not correct, and it would degrade the dictionary learning performance especially in the presence of noise [25]. However, when there is no information about the generation model, the existing algorithms such as K-SVD or MDU are good choices to perform the dictionary learning for sparse representation of the signals. In this paper, we consider a well-known dynamic model, especially in medical signals [32, 5, 1, 24], for the generation of the training signals. We assume that there are some hidden states which are activated under the first-order Markovian model with a fixed transition probability matrix. This means that there is a hidden Markov model (HMM). Each training signal ($\mathbf{y}_k$) is generated when the corresponding state is activated and each state has its own specific dictionary. Then, we explain how we should modify the dictionary learning algorithms and apply them on the considered dynamic model using Maximum Likelihood Estimator (MLE) and Expectation Maximization (EM). The considered scenario has been investigated in [36, 20] with some differences. In the following, we briefly explain them and state their differences with this study.

The authors of [36] classify the surgical gestures in robotic surgical tasks. The goal of gesture classification is assigning a surgeme label or state to each surgery trial ($\mathbf{y}_k$). In fact, the number of states is equal to the number of classes in this classification problem. The dependency among the surgery trials is modeled by a HMM, and each state has a specific dictionary which can sparsely represent the trials in that state. Since the problem is a classification task, the signals are divided to the training and testing sets. The surgeme labels are assumed to be known in the training phase. Hence, the transition probabilities can be directly computed from the frequency of surgeme transitions, and the dictionary of each state is learned using regular dictionary learning algorithms such as K-SVD. Then, the transition probabilities and the learned dictionaries are used for classification of the testing data. The main difference between the model considered in [36] and this study is regarding the labels of the signals. We assume that they are unknown which makes the problem much more difficult. This point is similar to the existing difference between the clustering and classification problem.

In [20], the dynamic of the dictionary learning problem is modeled by a HMM, and each state in the Markov chain is characterized by a set of active and inactive atoms. Since at most $N_0$ atoms are activated at each time instant, the number of states is considered as $\sum_{n=1}^{N_0} \binom{N}{n}$. Then, it is proposed to use the change-of-measure technique to define a new probability measure over the

set of observations. Finally, the parameters of the considered model are recursively computed. It is worth mentioning that since the states are unknown, the proposed method in [20] includes the EM trick similar to one employed in typical HMM problem. The states definition is the main difference between the scenario considered in [20] and this study.

The proposed approach in this study is not an independent dictionary learning algorithm. It is a general approach that can be applied on any existing dictionary learning algorithm to make it work better for statistically dependent training signals, where this dependency can be modeled by a HMM. It is worth mentioning that the initial idea of this study was published in the conference article [4]. In this study, we present a comprehensive treatment of the initial idea including 1) the accurate definition of the considered model with its mathematical reasons, 2) the explanation of the proposed approach with all of its important mathematical details, 3) the verification of the proposed approach, and the comparison of the proposed approach with state-of-the-art approaches using simulations in different scenarios, and 4) the presentation of a real application in the neural data set for the considered model.

### *1.4. Paper Organization and Notation*

The rest of this paper is organized as follows. In Section 2, we explain the considered model for the generation of signals, and present an example to show why the existing algorithms fail in retrieving the dictionary and the coefficient matrix from the signals generated under the considered model. The proposed approach for estimating the unknown parameters are explained in Section 3. Simulation results and a real application for the considered model are presented in Section 4, and finally, the discussion and concluding remarks are reported in Section 5.

We use bold small letters to show vectors ($\mathbf{a}$) and bold capital letters to show matrices ($\mathbf{A}$). The $j^{th}$ column and the $(i,j)^{th}$ entry of $\mathbf{A}$ are shown by $\mathbf{a}_j$ and $a_{ij}$, respectively. The superscript $(.)^T$ denotes vector or matrix transpose. $[\mathbf{a}_1\ \mathbf{a}_2]$ and $[\mathbf{a}_1\ ;\ \mathbf{a}_2]$ respectively show the horizontal and vertical concatenation of $\mathbf{a}_1$ and $\mathbf{a}_2$. The identity matrix is shown by $\mathbf{I}$, and finally, $\|\mathbf{A}\|_F$ and $\|\mathbf{a}\|_p$ show the Frobenius and $l_p$ norms, respectively.

## 2. Model Definition

We assume that there are a few ($S$) states and each signal is generated when the corresponding state is activated. The states are activated under HMM with a fixed transition probability matrix $\mathbf{P} \in \mathbb{R}^{S\times S}$. The $s^{th}$ state consists of $L$ atoms. When a signal corresponding to the $s^{th}$ state is generated, at most $N_0$ atoms are randomly selected from $L$ atoms, and participate in the generation of the signal. The considered model, when there are $S = 2$ states with $L = 2$ atoms in each state, and at most $N_0 = 1$ atom participates in generating the $M = 2$ dimensional signals, is schematically shown in Fig. 2. It is worth mentioning that $N = S \times L = 4$.

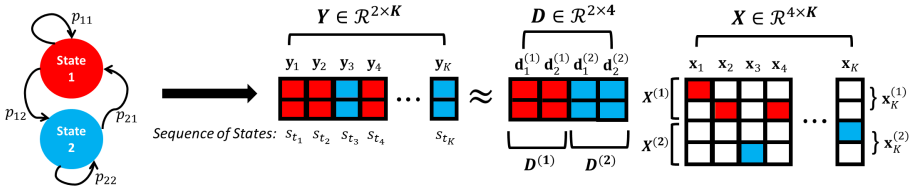


Figure 2: Considered model for the generation of signals, when $N = 4$, $M = S = L = 2$, and $N_0 = 1$.

The probability of transition from the $i^{th}$ state to the $j^{th}$ state is shown by $p_{ij}$, which is the $(i,j)^{th}$ entry of the transition probability matrix $\mathbf{P}$. The sequence of states is shown by $\{s_{t_1}, s_{t_2}, ..., s_{t_K}\}$ where $s_{t_k}$ shows the corresponding state for $\mathbf{y}_k$ (e.g., in Fig. 2, each $s_{t_k}$ is either 1 or 2), and it is only dependent to $s_{t_{k-1}}$ due to the presence of first-order Markovian dependency in HMM. The dictionary $\mathbf{D} = [\mathbf{D}^{(1)}\ \mathbf{D}^{(2)} ... \mathbf{D}^{(S)}]$ is divided into $S$ subdictionaries associated with different states, and $\mathbf{d}_l^{(s)} \in \mathbb{R}^M$ represents the $l^{th}$ atom from the $s^{th}$ state. The coefficient matrix $\mathbf{X} = [\mathbf{X}^{(1)}; \mathbf{X}^{(2)}...; \mathbf{X}^{(S)}]$ is also divided into $S$ submatrices associated with different states, and $\mathbf{x}_k^{(s)} \in \mathbb{R}^L$ shows the entries of the $k^{th}$ column of the coefficient matrix corresponding to the $s^{th}$ state. It should be noted that we can consider a separate dictionary and a separate coefficient matrix for each state, and do not mix the matrices as shown in Fig. 2. However, we consider the current decomposition because its form is similar to the typical dictionary learning problems.

The set of unknown parameters in the considered model is as follows:

$$\Omega = \underbrace{\{\mathbf{P}, \mathbf{D}, \mathbf{X}\}}_{\Theta} \cup \{s_{t_1}, s_{t_2}, ..., s_{t_K}\} \tag{8}$$

The noticeable point is that we assume the number of states ($S$), the number of atoms in each state ($L$), and the sparsity level ($N_0$) are fixed and known. Usually, there are three approaches employed for finding these parameters: 1) Sometimes, some prior information is available about the system and data. Hence, these parameters may exist in the prior knowledge. 2) Sometimes, these parameters can be found using the cross validation frameworks. For instance, the system is trained and tested by considering different $S$, $L$, and $N_0$. Then, the parameters which lead to the minimum training and testing errors are considered as the optimum values. 3) Sometimes, the results must be comprehensible. For instance, in neuroscience and intracranial EEG signals, where the dictionary learning algorithms are employed to find the spike and wave discharges, each column of the dictionary corresponds to a specific spike. The obtained spikes must have biophysiological interpretation. Hence, the dictionary is learned with different $S$, $L$, and $N_0$. Then, the parameters which lead to the results with the best biophysiological interpretation are considered as the optimum values [5].

It should also be noted that in the more general form of the considered model, the states can have common atoms, the number of atoms in each state can be

different, or the selection of the atoms in each state for generating the signals can be under some probability density functions. For simplicity, we ignore these scenarios, however, the same approach as proposed in Section 3 can be applied to obtain the model parameters in these scenarios.

Now, using a simple example, we show why the existing algorithms which use (3) as the sparsification step cannot be applied on the signals generated under the considered model.

**Example:** Assume that there are $S = 2$ states with $L = 1$ atom in each one, and at most $N_0 = 1$ atom participates in generating the $M = 2$ dimensional signals. Now, we are in the sparsification step, and we want to estimate the state of the observed signal $\mathbf{y}_k$, or in other words, select the best atom for $\mathbf{y}_k$. The scatter plot for $\mathbf{y}_{k-1}$ and $\mathbf{y}_k$, in the presence of additive noise is schematically shown in Fig. 3, and assume that $\mathbf{y}_{k-1}$ has been assigned to state 2, i.e., $s_{t_{k-1}} = 2$.

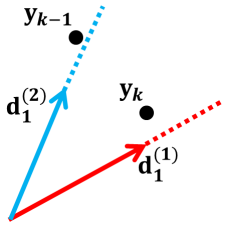


Figure 3: Scatter plot of two consecutive signals generated under HMM. Assigning $\mathbf{y}_k$ to one of the atoms is not independent form the activated state (or atom) for $\mathbf{y}_{k-1}$. $\mathbf{d}_1^{(s)}$ shows the atom associated with the $s^{th}$ state.

Since there is HMM in activation of the states, if we use the maximum aposteriori (MAP) estimator to find $s_{t_k}$, we get

$$\underbrace{p(s_{t_k} = 1 \,|\, \mathbf{y}_k, s_{t_{k-1}}, \mathbf{D})}_{f_1} \underset{1}{\overset{2}{\lessgtr}} \underbrace{p(s_{t_k} = 2 \,|\, \mathbf{y}_k, s_{t_{k-1}}, \mathbf{D})}_{f_2}. \tag{9}$$

According to Bayes' rule, the left and right sides of (9) can be simplified and written as

$$\begin{aligned} f_1 &= \frac{\overbrace{p(s_{t_k} = 1 \,|\, s_{t_{k-1}})}^{p_{21}} \, p(\mathbf{y}_k \,|\, s_{t_k} = 1, \mathbf{d}_1^{(1)})}{p(\mathbf{y}_k|\mathbf{D})}, \\ f_2 &= \frac{\overbrace{p(s_{t_k} = 2 \,|\, s_{t_{k-1}})}^{p_{22}} \, p(\mathbf{y}_k \,|\, s_{t_k} = 2, \mathbf{d}_1^{(2)})}{p(\mathbf{y}_k|\mathbf{D})}. \end{aligned} \tag{10}$$

It can be seen that not only are the conditional probabilities of $\mathbf{y}_k$ important for estimating $s_{t_k}$, but also the probabilities of transition between the states affect the decision making for $s_{t_k}$. Therefore, dividing the sparsification step

into $K$ different sparse recovery problems as performed in (3) is not correct in the considered model.

Three important points must be mentioned about the considered model. The first point is that the explanations provided in the aforementioned example can only be applied on the time-dependent sequences and cannot be generalized to any correlated signals. The second point is that based on Fig. 3 and (10), whenever the signal to noise ratio (SNR) increases, the produced error using the stated sparsification step in (3) decreases. For instance, if the signals were noise free, which is not a practical assumption, the dictionary could be learned without any error using the existing algorithms following the stated sparsification step in (3). Finally, the last point is that whenever the probabilities of transition among the states become close to the uniform distribution, i.e., the states become independent, the produced error using (3) would decrease. For instance, if $p_{21}$ and $p_{22}$ were equal to 0.5 in (10), they would not affect the decision making, and we would have the same sparsification step as in (3).

Now, the model definition is complete. In the following, we explain how to estimate the set of unknown parameters $\Omega = \Theta \cup \{s_{t_1}, s_{t_2}, ..., s_{t_K}\}$ from the observed signals $\mathbf{Y}$.

## 3. Proposed Approach

Since the activated state for each signal is unknown, we first estimate the probability of activation of each state for each signal, and then, find the correct state for each signal. Probability of activation of each state for each signal is a function of $\mathbf{Y}$ and $\Theta$. Therefore, at first we show how to estimate $\Theta$, and then, explain how to find the activated state for each signal, or in other words, find the sequence of states.

In the presence of noise, each signal $\mathbf{y}_k \in \mathbb{R}^M$ can be expressed as

$$\mathcal{H}_k^{(s)} : s_{t_k} = s \rightarrow \quad \mathbf{y}_k = \mathbf{D}^{(s)}\mathbf{x}_k^{(s)} + \mathbf{n}_k, \tag{11}$$

where $\mathcal{H}_k^{(s)}$ denotes the hypothesis that the $s^{th}$ state is active for generating $\mathbf{y}_k$. As explained before, $\mathbf{D}^{(s)} \in \mathbb{R}^{M \times L}$ and $\mathbf{x}_k^{(s)} \in \mathbb{R}^L$ represent the subdictionaries corresponding to the $s^{th}$ state, and the entries of the $k^{th}$ column of the coefficient matrix associated with the $s^{th}$ state, respectively. $\mathbf{n}_k \in \mathbb{R}^M$ shows the additive noise which is considered zero-mean white Gaussian with $\mathcal{N}(0, \sigma^2\mathbf{I})$ distribution. Hence, the conditional probability density function (pdf) of $\mathbf{y}_k$ can be expressed as

$$f(\mathbf{y}_k | s_{t_k} = s, \Theta) = (\frac{1}{\sqrt{2\pi\sigma^2}})^M \exp(-\frac{\|\mathbf{y}_k - \mathbf{D}^{(s)}\mathbf{x}_k^{(s)}\|_2^2}{2\sigma^2}). \tag{12}$$

We use the Maximum Log-likelihood Estimator (MLE) to find the set of un-

known parameters, i.e.,

$$\Theta = \underset{\Theta}{\text{argmax}} \quad log(f(\mathbf{Y}|\Theta))$$
$$s.t. \quad \|\mathbf{d}_l^{(s)}\|_2 = 1, \quad \|\mathbf{x}_k^{(s)}\|_0 \leq N_0$$
$$s = 1, 2, ..., S, \quad l = 1, 2, ..., L, \quad k = 1, 2, ..., K \tag{13}$$

Actually, the constraint $\|\mathbf{x}_k^{(s)}\|_0 \leq N_0$ must only be considered for the coefficients that are associated with the activated state for the $k^{th}$ signal, but we have considered it for all of the states. The reason is the fact that was explained at the beginning of this Section: Since we do not know the sequence of states beforehand, we try to find the probability of activation of each state for each signal, and so, these coefficients are estimated based on the probabilities. In the final step, when the sequence of states is determined, we modify the coefficient matrix such that only the coefficients corresponding to the activated states become non-zero.

Since the noise vectors $\mathbf{n}_k$ for $k = 1, 2, ..., K$ are independent, the stated objective function in (13) can be expressed as

$$\begin{aligned} log(f(\mathbf{Y}|\Theta)) &= \sum_{k=1}^{K} log(f(\mathbf{y}_k|\Theta)) \\ &= \sum_{k=1}^{K} log\{\sum_{s=1}^{S} p(s_{t_k} = s|\mathbf{Y}, \Theta) f(\mathbf{y}_k|s_{t_k} = s, \Theta)\}, \end{aligned} \tag{14}$$

where $p(s_{t_k} = s|\mathbf{Y}, \Theta)$ shows the probability of activation of the $s^{th}$ state in the generation of $\mathbf{y}_k$ given all of the observations and the model parameters. Since $p(s_{t_k} = s|\mathbf{Y}, \Theta)$ is unknown, and the summation over $s$ in (14) is like an expectation operator, we maximize (14) using Expectation-Maximization (EM) method [12]. Some feasible initializations are considered for the parameters, then, the following two steps (*E-step* and *M-step*) are alternately performed until convergence of the parameters.

Before explaining *E-step* and *M-step*, it is worth mentioning that feasible initializations means that we select positive values for the entries of the transition probability matrix ($\mathbf{P}$) such that sum of its entries in each row is equal to one. Then, we generate a sequence of states ($\{s_{t_1}, s_{t_2}, ..., s_{t_K}\}$) corresponding to the considered transition probability matrix. The MATLAB function *hmmgenerate* can be employed to do this. We also select random values for the dictionary ($\mathbf{D}$) and make its columns unit norm. Finally, corresponding to the activated state at each time instant, we choose random values for the coefficient matrix ($\mathbf{X}$) such that each column at most has $N_0$ non-zero entries.

### *3.1. Expectation Step (E-step)*

In this step, we assume that the main parameters of the model, i.e., the dictionary, the coefficient matrix and the transition probability matrix, are known

and fixed, and we estimate $p(s_{t_k} = s|\mathbf{Y}, \Theta)$ for $k = 1, 2, ..., K$ and $s = 1, 2, ..., S$. We use the forward-backward procedure to estimate $p(s_{t_k} = s|\mathbf{Y}, \Theta)$ as explained in Appendix A. It must be mentioned that in this procedure, $p(s_{t_k} = s, s_{t_{k+1}} = z|\mathbf{Y}, \Theta)$, i.e., the probability of being in state $s$ for the $k^{th}$ signal and being in state $z$ for the $(k+1)^{th}$ signal given the observations and the model parameters, is also calculated which would be used in the next step to estimate the transition probability matrix.

### *3.2. Maximization Step (M-step)*

In this step, we estimate $\Theta$ which consists of the transition probability matrix ($\mathbf{P}$), the dictionary ($\mathbf{D}$) and the coefficient matrix ($\mathbf{X}$), assuming $p(s_{t_k} = s|\mathbf{Y}, \Theta)$ for $k = 1, 2, ..., K$ and $s = 1, 2, ..., S$ are known and fixed.

At first, we estimate the transition probability matrix. The probability of transition from state $s$ to state $z$ ($p_{sz}$) which is the $(s, z)^{th}$ entry of $\mathbf{P}$, shows the expected number of transitions from state $s$ to state $z$ relative to the expected total number of transitions away from state $s$. Therefore, we can find $p_{sz}$ as follows:

$$[\mathbf{P}]_{sz} = p_{sz} \;\rightarrow\; p_{sz} = \frac{\sum_{k=1}^{K-1} p(s_{t_k} = s, s_{t_{k+1}} = z|\mathbf{Y}, \Theta)}{\sum_{k=1}^{K-1} p(s_{t_k} = s|\mathbf{Y}, \Theta)} \tag{15}$$

By substituting (32) and (33) (Appendix A) in (15), the entries of the transition probability matrix and consequently the transition probability matrix are estimated.

Now, we back to the proposed objective function in (14) to estimate the dictionary and the coefficient matrix. Direct maximization of (14) is difficult due to the presence of the summation operator inside the log term. Hence, we consider a lower bound of (14) using Jensen's inequality (log-concavity), and then, maximize it as follows:

$$log(f(\mathbf{Y}|\Theta)) \geq \overbrace{\sum_{k=1}^{K}\sum_{s=1}^{S} p(s_{t_k} = s|\mathbf{Y}, \Theta) log\{f(\mathbf{y}_k|s_{t_k} = s, \Theta)\}}^{h(\Theta)}$$

$$\Theta^* = \underset{\Theta}{\text{argmax}} \;\; h(\Theta) \tag{16}$$

By substituting (12) instead of $f(\mathbf{y}_k|s_{t_k} = s, \Theta)$ in (16), we get

$$h(\Theta) = b_0 - \frac{1}{2\sigma^2}\sum_{k=1}^{K}\sum_{s=1}^{S} \underbrace{p(s_{t_k} = s|\mathbf{Y}, \Theta)}_{w_{ks}} \|\mathbf{y}_k - \mathbf{D}^{(s)}\mathbf{x}_k^{(s)}\|_2^2, \tag{17}$$

where $b_0 = -\frac{MK}{2} log(2\pi\sigma^2)$ is a constant which is not related to the parameters, and $w_{ks} = p(s_{t_k} = s|\mathbf{Y}, \Theta)$ is known because it was estimated in *E-Step*. It can be seen that we can separate the parameters of each state in $h(\Theta)$. In fact,

maximizing $h(\Theta)$ is equivalent to solving the following independent optimization problems:

$$\{\mathbf{D}^{(s)}, \mathbf{X}^{(s)}\} = \underset{\mathbf{D}^{(s)}, \mathbf{X}^{(s)}}{\operatorname{argmin}} \quad \sum_{k=1}^{K} w_{ks} \|\mathbf{y}_k - \mathbf{D}^{(s)} \mathbf{x}_k^{(s)}\|_2^2$$
$$s = 1, 2, ..., S \tag{18}$$

According to (13), the following constraints must also be considered for each of the optimization problems:

$$\|\mathbf{d}_l^{(s)}\|_2 = 1, \quad l = 1, 2, ..., L$$
$$\|\mathbf{x}_k^{(s)}\|_0 \leq N_0, \quad k = 1, 2, ..., K \tag{19}$$

It can be seen that (18) and (19) form a constrained optimization problem which is very similar to (1). The only difference is the presence of the weight $w_{ks}$ in the new optimization problem. By defining the variables $\mathbf{y}_k^{'} = \sqrt{w_{ks}}\mathbf{y}_k$ and $\mathbf{x}_k^{(s)^{'}} = \sqrt{w_{ks}}\mathbf{x}_k^{(s)}$, the optimization problem stated in (18) converts to the following problem which is now completely similar to (1).

$$\{\mathbf{D}^{(s)}, \mathbf{X}^{(s)^{'}}\} = \underset{\mathbf{D}^{(s)}, \mathbf{X}^{(s)^{'}}}{\operatorname{argmin}} \quad \sum_{k=1}^{K} \|\mathbf{y}_k^{'} - \mathbf{D}^{(s)} \mathbf{x}_k^{(s)^{'}}\|_2^2$$
$$s = 1, 2, ..., S \tag{20}$$

Hence, the methods like MOD or K-SVD or any other dictionary learning method can be used to solve the new constrained optimization problem. It should be noted that by determination of the variable $\mathbf{x}_k^{(s)^{'}}$, the main variable $\mathbf{x}_k^{(s)}$ is also found ($\mathbf{x}_k^{(s)} = (\sqrt{w_{ks}})^{-1}\mathbf{x}_k^{(s)^{'}}$).

By determination of the subdictionaries ($\mathbf{D}^{(s)}$) and the corresponding coefficient matrices ($\mathbf{X}^{(s)}$) for $s = 1, 2, ..., S$, the *M-Step* is done.

By performing a few iterations between the *E-Step* and the *M-Step*, $\Theta$ is estimated. By determination of $\Theta$, we can use Viterbi algorithm [16, 37] to find the activated state for each signal, or in other words, find the sequence of states $\{s_{t_1}, s_{t_2}, ..., s_{t_K}\}$. It should be noted that we cannot only use (12) to find the activated state at each time instant. We should find the optimal sequence of states by considering all of the observations because there is a time-dependent sequence.

The noticeable point is that after determination of the sequence of states, a minor modification must be applied on the obtained coefficient matrix as mentioned at the beginning of Section 3. In each column of the coefficient matrix ($\mathbf{x}_k$), we keep the coefficients associated with the activated state ($\mathbf{x}_k^{(s_k)}$), and make the other coefficients equal to zero. This is because for each signal, only the atoms associated with the activated state participate in generating the signal. Here, the explanation of the proposed approach is complete. The proposed approach is briefly explained in Algorithm 1.

**Algorithm 1** Proposed approach for learning dictionary from signals with HMM dependency.

**Goal:** perform the factorization $\mathbf{Y} = \mathbf{D}\,\mathbf{X}$ and find the unknown parameters:

$$\Omega = \underbrace{\{\mathbf{P}, \mathbf{D}, \mathbf{X}\}}_{\Theta} \cup \{s_{t_1}, s_{t_2}, ..., s_{t_K}\}$$

**Proposed Approach:** perform *E-step* and *M-step* alternately as follows:
**while** *the convergence criterion is not satisfied* **do**

***E-step*:**

1. calculate $\alpha_k(s)$ and $\beta_k(s)$ according to (27) and (29).
2. calculate $w_{ks} = p(s_{t_k} = s|\mathbf{Y}, \Theta)$ according to (32).
3. calculate $p(s_{t_k} = s, s_{k+1} = z|\mathbf{Y}, \Theta)$ according to (33).

***M-step*:**

1. estimate the transition probability matrix according to (15).
2. **for** *s=1,2,...,S* **do**

$$\{\mathbf{D}^{(s)}, \mathbf{X}^{(s)}\} = \underset{\mathbf{D}^{(s)}, \mathbf{X}^{(s)}}{\operatorname{argmin}} \quad \sum_{k=1}^{K} w_{ks} \|\mathbf{y}_k - \mathbf{D}^{(s)}\mathbf{x}_k^{(s)}\|_2^2$$
$$s.t. \quad \|\mathbf{d}_l^{(s)}\|_2 = 1, \qquad l = 1, 2, ..., L$$
$$\|\mathbf{x}_k^{(s)}\|_0 \leq N_0, \quad k = 1, 2, ..., K$$

**end**

**end**

**Finally:** find the sequence of states using Viterbi algorithm and modify $\mathbf{X}$ according to the obtained sequence of states.

We must also mention another noticeable point at the end of this section. There is an important difference between parameters estimation of typical HMM [32] and the model considered in this paper. In typical HMM, assuming that the conditional pdf of the observations is Gaussian in each state, the main parameters of the model are the average and the covariance of the signals in each state, which are static parameters because they are fixed over time. While in the model considered in this manuscript, the dictionary and the coefficient matrix are the main parameters in each state. The dictionary is the static parameter because it is fixed over time but each column of the coefficient matrix is the dynamic parameter because it changes over time. Although there are similarities in parameter estimation of typical HMM and the model considered in this manuscript, the concepts are completely different.

## 4. Experimental Results

In this Section, we first apply the proposed approach on some of the state-of-the-art algorithms, and compare their performance using simulated data. Then,

we present a real application for the considered model.

### *4.1. Recovery of a known dictionary*

As explained earlier, optimizing (18) can be done using any dictionary learning algorithm. We use MOD [15], K-SVD [3], SimCo [11], and MDU [35] for this optimization problem, and call the algorithms as "New MOD", "New K-SVD", "New SimCo", and "New MDU", respectively. For all of the methods, OMP is employed when a sparse recovery problem is to be solved. It is emphasized that the goal of all algorithms is recovering a dictionary with $N = L \times S$ atoms.

Moreover, we use the successful recovery rate criterion, as mentioned in [3], to compare the algorithms, which is the number of correctly estimated atoms divided by the number of atoms. We say that an atom is correctly estimated if its correlation with the actual atom was above 0.99.

**Data generation:** We generate an overcomplete dictionary by a random matrix of size $15 \times 50$ ($M = 15$, $N = 50$) with zero-mean and unit-variance independent and identically distributed (i.i.d.) Gaussian entries, followed by normalizing the columns. We assume that there are $S = 2$ states which are activated under the HMM with the transition probability matrix $\mathbf{P} \in \mathbb{R}^{2\times 2}$, and each state consists of $L = 25$ atoms. Therefore, the first part of the dictionary $\mathbf{D}^{(1)} \in \mathbb{R}^{15\times 25}$ is assigned to the first state and the second part $\mathbf{D}^{(2)} \in \mathbb{R}^{15\times 25}$ is assigned to the second state. For generating the coefficient matrix $\mathbf{X} \in \mathbb{R}^{50\times 2000}$ ($N = 50$, $K = 2000$), a sequence of sates $\{s_{t_1}, s_{t_2}, ..., s_{t_{2000}}\}$ is generated under the considered model. The MATLAB function *hmmgenerate* can be used in this step. For each signal $\mathbf{x}_k \in \mathbb{R}^{50}$, we randomly select $N_0$ entries from $\mathbf{x}_k^{(s_{t_k})} \in \mathbb{R}^{25}$ as the non-zero entries. Then, the values of non-zero entries are chosen from Gaussian random variables with zero-mean and unit variance. Finally, the signals $\mathbf{y}_k \in \mathbb{R}^{15}$ for $k = 1, 2, ..., K$ are generated as

$$\mathbf{y}_k = \mathbf{D}\mathbf{x}_k + \mathbf{n}_k \tag{21}$$

where $\mathbf{n}_k \in \mathbb{R}^{15}$ is the additive noise which is considered zero-mean white Gaussian with $\mathcal{N}(0, \sigma^2\mathbf{I})$ distribution. We adjust $\sigma^2$ to achieve the desired $\text{SNR}_{dB}$ which is defined as follows [34]:

$$\text{SNR}_{dB} = 10\, log(\frac{1}{K}\sum_{k=1}^{K}\frac{\|\mathbf{y}_k\|_2^2}{\|\mathbf{n}_k\|_2^2}) \tag{22}$$

Two different scenarios are considered for the transition probability matrix. In the first scenario, we assume that the entries of $\mathbf{P}$ are almost uniformly distributed, meaning that the states are independent. In this scenario, the transition probability matrix is considered as

$$\mathbf{P}_1 = \begin{bmatrix} 0.55 & 0.45 \\ 0.45 & 0.55 \end{bmatrix} \tag{23}$$

In the second scenario, we assume that the states are highly dependent, and consider the transition probability matrix as

$$\mathbf{P}_2 = \begin{bmatrix} 0.95 & 0.05 \\ 0.10 & 0.90 \end{bmatrix} \tag{24}$$

**Results for the first scenario (independent states):** The average percentage of the successful recovery rate over 100 trials in different $\text{SNR}_{dB}$ and $N_0$ is reported in Table 1. The values reported in parentheses are corresponding to the results obtained from the new versions of the algorithms.

Table 1: Percentage of successful recovery rate in the first scenario where the states are activated almost independently from each other. The values reported in parentheses are corresponding to the results obtained from the new versions of the algorithms.

| $\text{SNR}_{dB}$ | Method | $N_0 = 3$ | $N_0 = 4$ | $N_0 = 5$ |
|---|---|---|---|---|
| 10 | MOD | 78.5 (79.1) | 70.6 (71.7) | 2.6 (2.9) |
| | K-SVD | 80.5 (81.4) | 78.8 (80.1) | 10.5 (12.6) |
| | SimCo | 83.6 (84.2) | 81.8 (82.5) | 18.2 (20.6) |
| | MDU | 84.4 (85.1) | 82.9 (83.7) | 23.3 (24.2) |
| 20 | MOD | 84.6 (84.8) | 81.4 (83.7) | 74.7 (75.4) |
| | K-SVD | 86.6 (86.8) | 85.9 (86.1) | 81.3 (80.3) |
| | SimCo | 88.4 (89.1) | 87.6 (87.9) | 83.2 (84.4) |
| | MDU | 88.9 (91.4) | 87.7 (90.4) | 82.5 (82.9) |
| 30 | MOD | 86.5 (87.5) | 84.4 (86.3) | 81.6 (83.4) |
| | K-SVD | 88.8 (89.9) | 87.1 (88.6) | 84.3 (84.5) |
| | SimCo | 89.4 (90.2) | 88.5 (89.6) | 86.3 (86.8) |
| | MDU | 89.7 (90.3) | 89.1 (89.9) | 85.4 (86.7) |
| 100 | MOD | 88.2 (88.8) | 86.3 (86.8) | 83.5 (84.7) |
| | K-SVD | 89.2 (90.6) | 88.5 (89.2) | 87.5 (87.4) |
| | SimCo | 90.5 (91.2) | 89.1 (89.6) | 88.1 (89.4) |
| | MDU | 91.1 (91.2) | 89.7 (90.2) | 88.6 (89.3) |

As reported, the results obtained from the dictionary learning algorithms are similar to ones obtained from the new versions of the algorithms. The reason is that the states are activated independently from each other, and breaking the sparsification step to $K$ sparse recovery problems is a correct strategy.

**Results for the second scenario (dependent states):** The average percentage of the successful recovery rate over 100 trials in different $\text{SNR}_{dB}$ and $N_0$ is reported in Table 2.

As reported, the new versions of the algorithms perform better than the initial versions especially in low SNRs, which is in accordance with the remarks stated in the end of Section 2. The reason is that the states are highly dependent and solving the sparse recovery problem for each signal independently from other signals is not a correct strategy. To clarify this point, a part of the obtained sequence of states in a specific trial after applying MOD and New MOD are shown in Fig. 4. In the considered trial, the parameters are $N_0 = 3$ and

$\text{SNR}_{dB} = 20$. As shown, the states in two of the samples are wrongly estimated by MOD, while these errors do not exist in the sequence of states obtained by New MOD.

Table 2: Percentage of successful recovery rate in the second scenario where the states are dependent. The values reported in parentheses are corresponding to the results obtained from the new versions of the algorithms.

| $\text{SNR}_{dB}$ | Method | $N_0 = 3$ | $N_0 = 4$ | $N_0 = 5$ |
|---|---|---|---|---|
| 10 | MOD | 67.3 (79.2) | 60.7 (69.3) | $\simeq 0$ (2.9) |
| | K-SVD | 70.9 (80.8) | 67.9 (79.2) | 2.4 (12.1) |
| | SimCo | 74.2 (82.8) | 69.4 (82.3) | 13.6 (21.7) |
| | MDU | 75.2 (83.9) | 70.4 (83.5) | 16.2 (24.9) |
| 20 | MOD | 72.7 (85.2) | 66.2 (82.8) | 48.3 (76.1) |
| | K-SVD | 78.2 (85.9) | 73.8 (84.2) | 70.6 (79.3) |
| | SimCo | 79.1 (87.6) | 75.2 (85.3) | 71.4 (80.7) |
| | MDU | 80.6 (89.8) | 77.8 (86.2) | 73.4 (79.3) |
| 30 | MOD | 83.5 (86.4) | 82.6 (85.6) | 79.7 (84.4) |
| | K-SVD | 87.2 (89.1) | 85.6 (89.2) | 81.8 (83.3) |
| | SimCo | 87.8 (89.2) | 87.1 (89.4) | 82.9 (84.6) |
| | MDU | 89.4 (90.7) | 88.2 (89.8) | 83.7 (84.1) |
| 100 | MOD | 86.3 (87.5) | 85.7 (85.6) | 83.2 (83.4) |
| | K-SVD | 89.6 (90.1) | 88.3 (90.2) | 87.1 (87.7) |
| | SimCo | 90.2 (91.8) | 89.5 (91.6) | 87.2 (87.8) |
| | MDU | 91.3 (91.6) | 89.5 (91.9) | 88.4 (88.9) |

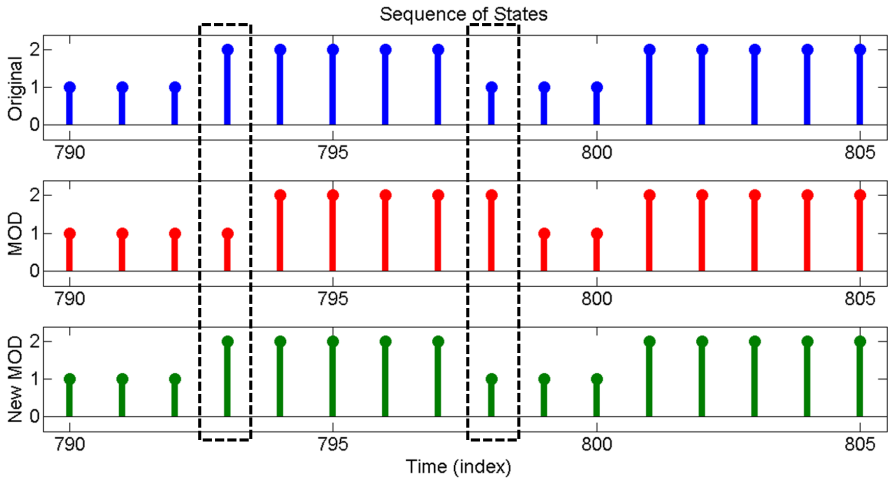


Figure 4: The obtained sequence of states after applying MOD and New MOD on a specific trial in which the parameters are $N_0 = 3$ and $\text{SNR}_{dB} = 20$. The dashed rectangles specify the time points in which the corresponding states are not correctly estimated by MOD.

It must be mentioned that the new versions of the algorithms have higher computational complexity than their original versions because the *E-step* must be performed in the proposed approach which makes the problem much more complex. If we examine the computations involved in performing the *E-step* which consists of the forward and backward procedures, it can be shown that we almost require $S^2K$ calculations, or more preciesly, $S(S+1)(K-1)+S$ multiplications and $S(S-1)(K-1)$ additions in each iteration. Hence, the new version of the algorithms have $O(S^2K)$ additional computational complexity per iteration.

To show the computational complexity quantitatively, the average convergence time of the algorithms in the second scenario when $\text{SNR}_{dB} = 20$ is reported in Table 3. Although the convergence time is a rough criterion to show the computational complexity, it still can be found that the new versions of the algorithms perform slower than their original versions. The simulations were performed on a system with 3.6 GHz CPU, 16 GB RAM, in MATLAB R2016a, and under Microsoft Windows 10 operating system.

Table 3: The average time of convergence (in seconds). The values reported in parentheses are corresponding to the results obtained from the new versions of the algorithms.

| Method | $N_0 = 3$ | $N_0 = 4$ | $N_0 = 5$ |
|---|---|---|---|
| MOD | 2.3 (5.8) | 5.1 (8.2) | 11.3 (24.2) |
| K-SVD | 14.6 (23.4) | 19.1 (32.5) | 25.8 (46.1) |
| SimCo | 4.6 (8.3) | 9.2 (18.2) | 17.1 (29.3) |
| MDU | 18.8 (31.2) | 43.5 (78.9) | 70.4 (131.6) |

### *4.2. Real Application*

Absence epilepsy is one of the several kinds of epilepsy which is more common in children. It is accompanied by sudden appearance of seizures in EEG recordings of patients suffering from this syndrome [22]. Many researchers have tried to understand the origin of seizures in the brain. Some of them pointed to thalamus, and some others considered somatosensory cortex as the origin of seizures [23, 29].

The most recent theory filling the gap between the mentioned origins claims that 1) a circuit between thalamus and somatosensory cortex generate the seizures, and 2) somatosensory cortex is the dominant active region during the seizures [38]. We aim to confirm the correctness of the aforementioned theory about the generation process of seizures using the proposed approach and a neural data set, which has been recently acquired by implanting an electrode in somatosensory cortex of an epileptic rat. The sampling frequency was $f_s = 2\,\text{kHz}$, and all the experiments have been submitted and approved by local Ethical Committee and European Union guidelines (directive 86/609/EEC). The details about the data acquisition can be found in [18], and a part of the recorded seizures is shown in Fig. 5.

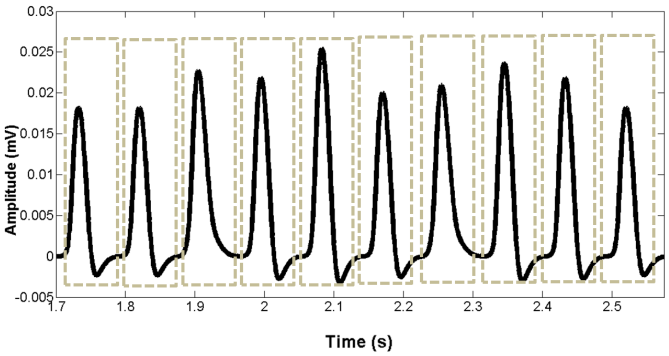


Figure 5: Recorded potentials during seizures. Spikes are specified by dashed rectangle.

The recorded data consist of few seizures, and each seizure is a train of spikes. We model the generation of spikes existing in a seizure. Based on the results obtained in [5], we assume that there are a few hidden states under HMM during a recorded seizure, and each spike of the recorded seizure, which is shown by dashed-rectangle in Fig. 5, is generated when the corresponding state is activated. Each state consists of a few specific spiky atoms which are linearly combined and construct the spikes. We aim to retrieve the states, their spiky atoms, and the probability of transition between the states from the recorded seizure using the proposed approach. We use the modified version of K-SVD to estimate the parameters.

For this purpose, we detect and align the spikes following the proposed methods in [31] and [7]. We consider each spike as one signal, and stack it in $\mathbf{y}_k \in \mathbb{R}^M$. Each spike consists of 175 samples, however, the noticeable point is that we can decrease the dimension of the signals using PCA to $M = 3$ with almost no loss because the shapes of the spikes are very similar. Hence, we first apply the PCA on the signals, and then, employ the proposed method to retrieve the model parameters. Finally, we come back to the actual dimension.

Since there is no prior information about the number of states ($S$), the number of spiky atoms ($L$) in each state, and the number of active atoms for each signal ($N_0$), we apply the proposed approach on the recorded seizures with different $S$, $L$, and $N_0$, then, we select the one which has the best biophysiological interpretation (e.g., the obtained spiky atoms must be smooth). The ranges of the parameters are considered as $1 \leq S \leq 3$, $1 \leq L \leq 2$, and $1 \leq N_0 \leq 2$. In the recorded seizures, the best results are obtained by considering $S = L = N_0 = 2$. The schematic diagram of the considered model for the recorded seizures is shown in Fig. 6 when $S = L = N_0 = 2$.

The obtained spiky atoms and transition probability matrix from one of the recorded seizures which has $K = 190$ spikes are shown in Fig. 7. It is worth mentioning that the spiky atoms are shown with their actual dimensions. Moreover, when we apply the proposed approach on the considered seizure, the parameters converge after almost 45 iterations between *E-step* and *M-step*.

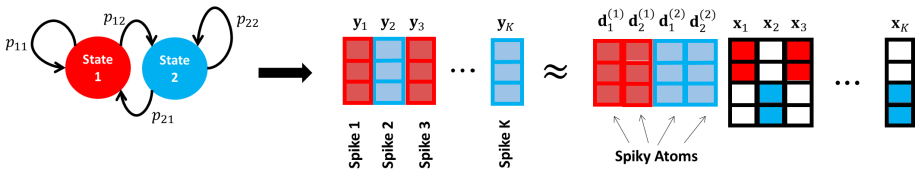


Figure 6: Considered model for the generation of spikes during recorded seizures.

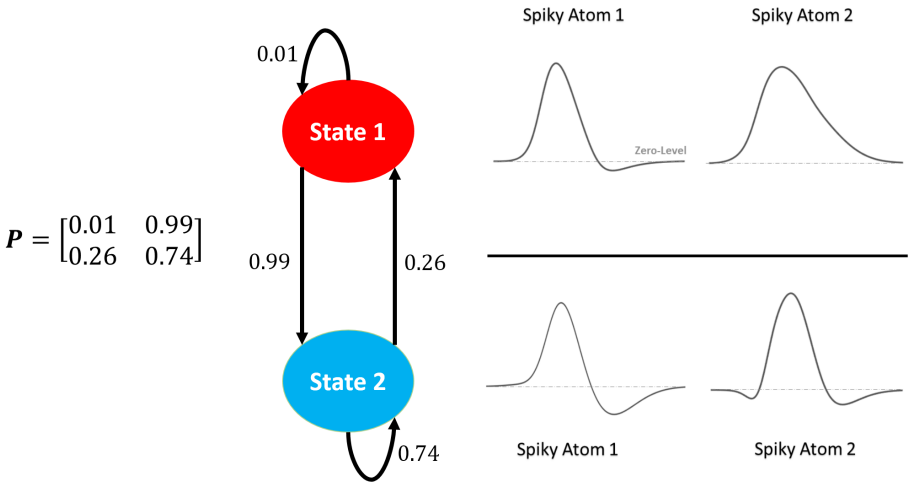


Figure 7: The obtained transition probability matrix and spiky atoms from the recorded data.

Each spiky atom is corresponded to a specific epileptic activity. By investigateing the obtianed epileptic activities, it can be shown that the obtained epileptic activities in the first state originate from thalamus, and the obtained epileptic activities in the second state are drived by somatosensory cortex [28]. In fact, the states are respectively corresponded to thalamus and somatosensory cortex, and a hidden Markov circuit between these states generates the seizures. Hence, the first part of the theory is proved.

According to the obtained transition probability matrix, the first state is unstable because $p_{11}$ is very small, and the second state is dominant because $p_{12}$ and $p_{22}$ are respectively larger than $p_{11}$ and $p_{21}$. For instance, the sequence of states for the part of the seizure, illustrated in Fig. 5, is shown in Fig. 8.

As shown here, the first state is unstable, and the second state is dominant during the seizure. The obtained result shows that the origin of the spiky atoms in the second state is more active than the ones in the first state during the seizures. Therefore, somatosensory cortex, which is corresponded to the second state, is the dominant active region during the seizures, and hence, the second part of the theory is also verified.

The final noticeable point is that since the states are highly dependent in the recorded seizures, it was necessary to employ the modified version of the

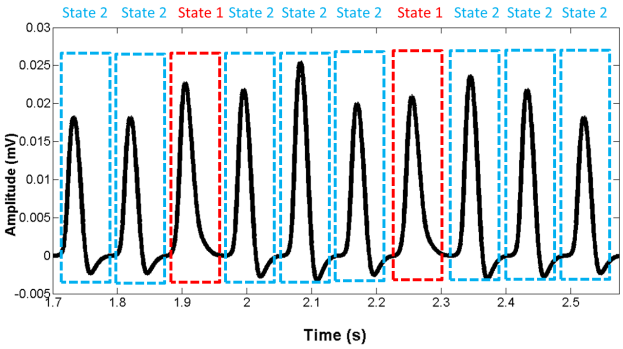


Figure 8: Sequence of states for a part of recorded seizure.

dictionary learning algorithms to estimate the parameters from the recorded seizures. We show the correctness of this statement using a cross validation framework. We compare the performance of the modified version of K-SVD and typical K-SVD when they are applied on the recorded seizures. Since there is no information about the actual values of the parameters and the problem is blind, the best criterion for comparing the algorithms is the generalization of the results obtained in different seizure. This means that we expect to obtain similar results when the algorithms are applied on different seizures. Hence, we consider two of the seizures which consists of $K_1 = 190$ and $K_2 = 213$ spikes. The first seizure is the one which was employed to show the results. Then, we apply K-SVD and New K-SVD on these seizures and obtain the spiky atoms. Finally, for each algorithm individually, we calculate the cross correlation coefficient between the obtained spiky atoms in these two seizures . The obtained results are reported in Table . 4. The values reported in parentheses are corresponding to the results obtained from the new version of K-SVD.

Table 4: The cross correlation coefficient between the obtained spiky atoms from two seizures which respectively have $K_1 = 190$ and $K_2 = 213$ spikes by applying "K-SVD" and "New K-SVD". The values reported in parentheses are corresponding to the results obtained from the new version of K-SVD.

| First Seizure | Second Seizure | | | |
|---|---|---|---|---|
| | $\mathbf{d}_1^{(1)}$ | $\mathbf{d}_2^{(1)}$ | $\mathbf{d}_1^{(2)}$ | $\mathbf{d}_2^{(2)}$ |
| $\mathbf{d}_1^{(1)}$ | 0.87 (0.95) | 0.84 (0.82) | 0.77 (0.93) | 0.65 (0.74) |
| $\mathbf{d}_2^{(1)}$ | 0.73 (0.80) | 0.86 (0.98) | 0.54 (0.80) | 0.73 (0.83) |
| $\mathbf{d}_1^{(2)}$ | 0.83 (0.93) | 0.78 (0.83) | 0.82 (0.97) | 0.84 (0.73) |
| $\mathbf{d}_2^{(2)}$ | 0.64 (0.72) | 0.75 (0.86) | 0.76 (0.72) | 0.75 (0.97) |

As reported, when the modified version of K-SVD is applied on the seizures, the results obtained from two seizures are more similar. In fact, the diagonal entries of the table, which should ideally be equal to one, confirm this statement.

It is worth mentioning that the considered validation framework is also a

good criterion for finding the hyperparameters such as $S$, $L$ and $N_0$. In fact, the set of parameters which leads to the best generalization of the results can be considered as the optimum hyperparameters.

## 5. Conclusion and Future Work

In several applications, such as those explained in this paper, dynamic models generate signals and hence, the generated signals are not statistically independent. The performance of the current dictionary learning algorithms degrades when they are applied on such dependent training signals. In this study, we investigated the dictionary learning problem for sparse representation of the signals when the training signals were generated based on a HMM. We showed that the dependency among the signals affects the selection of the proper atoms for each signal. Then, we proposed an approach using MAP and EM to improve the performance of the dictionary learning algorithms in the mentioned scenario. The proposed approach is not an independent dictionary learning algorithm. It is a general approach that we can employ for existing dictionary learning algorithms to make them work better for training signals with HMM dependency. Simulation results showed the effectiveness of the proposed approach especially when the signals were highly dependent and the SNR was low. We also presented a real application in medical signals for the considered model. For the future work, it would be interesting to consider other dynamic models for dependency of the signals, and perform the dictionary learning for sparse representation of the signals generated under the considered scenario. Many dynamic models such as auto-regressive (AR) models and higher order Markovian models can be considered for generation of the signals. Depending on the considered model, the approach used to modify the dictionary learning algorithms would completely change. Hence, we concentrated on a specific dynamic model in this paper, which can motivate the researchers to investigate the dictionary learning problem when there are other dynamic models in generation of signals.

## Bibliography


[1] Afzali, M., Fatemizadeh, E., Soltanian-Zadeh, H., 2017. Sparse registration of diffusion weighted images. Computer Methods and Programs in Biomedicine, in press .

[2] Afzali, M., Ghaffari, A., Fatemizadeh, E., Soltanian-Zadeh, H., 2016. Medical image registration using sparse coding of image patches. Computers in biology and medicine 73, 56–70.

[3] Aharon, M., Elad, M., Bruckstein, A., 2006. k -svd: An algorithm for designing overcomplete dictionaries for sparse representation. IEEE Transactions on Signal Processing 54, 4311–4322.

[4] Akhavan, S., Esmaeili, S., Babaie-Zadeh, M., Soltanian-Zadeh, H., 2018. Learning overcomplete dictionaries from markovian data, in: 2018 IEEE 10th Sensor Array and Multichannel Signal Processing Workshop (SAM), pp. 218–222.

[5] Akhavan, S., Phlypo, R., Soltanian-Zadeh, H., Studer, F., Depaulis, A., Jutten, C., 2017. Characterizing absence epileptic seizures from depth cortical measurements, in: 2017 25th European Signal Processing Conference (EUSIPCO), pp. 444–448.

[6] Barthélemy, Q., Larue, A., Mars, J.I., 2014. Decomposition and dictionary learning for 3d trajectories. Signal Processing 98, 423–437.

[7] Cabasson, A., Meste, O., 2008. Time delay estimation: A new insight into the woody's method. IEEE Signal Processing Letters 15, 573–576.

[8] Chandrasekhar, A., Natarajan, K., Yavarimanesh, M., Mukkamala, R., 2018. An iphone application for blood pressure monitoring via the oscillometric finger pressing method. Scientific reports 8, 13136.

[9] Chen, S.S., Donoho, D.L., Saunders, M.A., 1998. Atomic decomposition by basis pursuit. SIAM Journal on Scientific Computing 20, 33–61.

[10] Comon, P., Jutten, C., 2010. Handbook of Blind Source Separation: Independent Component Analysis and Applications. Elsevier.

[11] Dai, W., Xu, T., Wang, W., 2012. Simultaneous codeword optimization (simco) for dictionary update and learning. IEEE Transactions on Signal Processing 60, 6340–6353.

[12] Dempster, A.P., Laird, N.M., Rubin, D.B., 1977. Maximum likelihood from incomplete data via the em algorithm. Journal of the Royal Statistical Society: Series B (Methodological) 39, 1–22.

[13] El Mourabit, I., El Rhabi, M., Hakim, A., Laghrib, A., Moreau, E., 2017. A new denoising model for multi-frame super-resolution image reconstruction. Signal Processing 132, 51–65.

[14] Elad, M., 2010. Sparse and Redundant Representations. Springer, Berlin, Germany.

[15] Engan, K., Aase, S.O., Husoy, J.H., 1999. Method of optimal directions for frame design, in: Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on, pp. 2443–2446 vol.5.

[16] Forney, G.D., 1973. The viterbi algorithm. Proceedings of the IEEE 61, 268–278.

[17] Honarvar, B., Paramesran, R., Lim, C.L., 2014. Image reconstruction from a complete set of geometric and complex moments. Signal Processing 98, 224–232.

[18] Jarre, G., Altwegg-Boussac, T., Williams, M.S., Studer, F., Chipaux, M., David, O., Charpier, S., Depaulis, A., Mahon, S., Guillemain, I., 2017. Building up absence seizures in the somatosensory cortex: From network to cellular epileptogenic processes. Cerebral Cortex 27, 4607–4623.

[19] Jiang, R., Qiao, H., Zhang, B., 2016. Efficient fisher discrimination dictionary learning. Signal Processing 128, 28–39.

[20] Li, L., Scaglione, A., 2013. Learning hidden markov sparse models, in: 2013 Information Theory and Applications Workshop (ITA), pp. 1–10.

[21] Mairal, J., Bach, F., Ponce, J., Sapiro, G., 2009. Online dictionary learning for sparse coding, in: Proceedings of the 26th annual international conference on machine learning, ACM. pp. 689–696.

[22] Marten, F., Rodrigues, S., Suffczynski, P., Richardson, M.P., Terry, J.R., 2009. Derivation and analysis of an ordinary differential equation mean-field model for studying clinically recorded epilepsy dynamics. Phys. Rev. E 79, 021911.

[23] Meeren, H.K.M., Pijn, J.P.M., Coenen, A.M.L., Lopes Da Silva, O.H., 2002. Cortical Focus Drives Widespread Corticothalamic Networks During Spontaneous Absence Seizures in Rats. J. Neurosci 22, 1480–1495.

[24] Mohammadi-Nejad, A.R., Hossein-Zadeh, G.A., Soltanian-Zadeh, H., 2017. Structured and sparse canonical correlation analysis as a brain-wide multi-modal data fusion approach. IEEE Transactions on Medical Imaging 36, 1438–1448.

[25] Naderahmadian, Y., Beheshti, S., Tinati, M.A., 2016a. Correlation based online dictionary learning algorithm. IEEE Transactions on Signal Processing 64, 592–602.

[26] Naderahmadian, Y., Tinati, M.A., Beheshti, S., 2016b. Generalized adaptive weighted recursive least squares dictionary learning. Signal Processing 118, 89–96.

[27] Pati, Y.C., Rezaiifar, R., Krishnaprasad, P.S., 1993. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition, in: Signals, Systems and Computers, 1993. 1993 Conference Record of The Twenty-Seventh Asilomar Conference on, IEEE. pp. 40–44.

[28] Polack, P.O., 2016. Sensory processing during absence seizures. The Journal of physiology 594, 6439–6440.

[29] Polack, P.O., Guillemain, I., Hu, E., Deransart, C., Depaulis, A., Charpier, S., 2007. Deep Layer Somatosensory Cortical Neurons Initiate Spike-and-Wave Discharges in a Genetic Model of Absence Seizures. The Journal of Neuroscience 27, 6590–6599.

[30] Qi, J., Chen, W., 2018. Learning a discriminative dictionary for classification with outliers. Signal Processing 152, 255–264.

[31] Quiroga, R.Q., Nadasdy, Z., Ben-Shaul, Y., 2004. Unsupervised Spike Detection And Sorting With Wavelets And Superparamagnetic Clustering. Neural Comput 16, 1661–1687.

[32] Rabiner, L.R., 1989. A tutorial on hidden markov models and selected applications in speech recognition. Proceedings of the IEEE 77, 257–286.

[33] Shirzadian Gilan, M., Yavari Manesh, M., Mohammadi, A., 2016. Level crossing rate and average fade duration of amplify and forward relay channels with cochannel interference, in: European Wireless 2016; 22th European Wireless Conference, pp. 1–5.

[34] Skretting, K., Engan, K., 2010. Recursive least squares dictionary learning algorithm. IEEE Transactions on Signal Processing 58, 2121–2130.

[35] Smith, L.N., Elad, M., 2013. Improving dictionary learning: Multiple dictionary updates and coefficient reuse. IEEE Signal Processing Letters 20, 79–82.

[36] Tao, L., Elhamifar, E., Khudanpur, S., Hager, G.D., Vidal, R., 2012. Sparse hidden markov models for surgical gesture classification and skill evaluation, in: International conference on information processing in computer-assisted interventions, Springer. pp. 167–177.

[37] Viterbi, A., 1967. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. IEEE Transactions on Information Theory 13, 260–269.

[38] Williams, M.S., Altwegg-Boussac, T., Chavez, M., Lecas, S., Mahon, S., Charpier, S., 2016. Integrative properties and transfer function of cortical neurons initiating absence seizures in a rat genetic model. The Journal of physiology 594, 6733–6751.

[39] Yang, L., Fang, J., Cheng, H., Li, H., 2017. Sparse bayesian dictionary learning with a gaussian hierarchical model. Signal Processing 130, 93–104.

[40] Yavari Manesh, M., Olfat, A., 2017. Sigmoid function detector in the presence of heavy-tailed noise for multiple antenna cognitive radio networks, in: 2017 IEEE International Conference on Communications (ICC), pp. 1–5.

[41] Zhang, F., Cen, Y., Zhao, R., Hu, S., Mladenovic, V., 2018. Multi-separable dictionary learning. Signal Processing 143, 354–363.

# Appendices

## A. Forward-Backward Procedure For Estimating The Probabilities

### *A.1. Forward Procedure*

The forward variable is defined as follows:

$$\alpha_k(s) = p(\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_k, s_{t_k} = s \,|\, \Theta) \tag{25}$$

which shows the probability of observing the signals until the $k^{th}$ signal and observing state $s$ for the $k^{th}$ signal given the model parameters. The forward variable when $k = 1$ ($\alpha_1(s)$) is equal to

$$\alpha_1(s) = \underbrace{p(s_{t_1} = s)}_{\frac{1}{S}} f(\mathbf{y}_1 | s_{t_1} = s, \Theta). \tag{26}$$

We assume that the probabilities of activation of the states for the first signal are equal, hence, $p(s_{t_1} = s) = \frac{1}{S}$. Moreover, $f(\mathbf{y}_1 | s_{t_1} = s, \Theta)$ is known according to (12) because the model parameters are known in *E-step*. Therefore, $\alpha_1(s)$ is easily calculated. It can be shown that we can use the following recursive expression to calculate $\alpha_k(s)$:

$$\alpha_k(s) = [\sum_{q=1}^{S} \alpha_{k-1}(q) p_{qs}] f(\mathbf{y}_k | s_{t_k} = s, \Theta) \tag{27}$$

where $p_{qs}$ shows the $(q, s)^{th}$ entry of the transition probability matrix which is known in this step. Therefore, $\alpha_k(s)$ for $k = 1, 2, ..., K$ and $s = 1, 2, ..., S$ can be easily calculated using (12), (26) and (27).

### *A.2. Backward Procedure*

In a similar manner, we define the backward variable as follows:

$$\beta_k(s) = p(\mathbf{y}_{k+1}, \mathbf{y}_{k+2}, ..., \mathbf{y}_K \,|\, s_{t_k} = s, \, \Theta) \tag{28}$$

which shows the probability of observing signals from $(k+1)^{th}$ signal until the last signal, given state $s$ for the $k^{th}$ signal and given the model parameters. Assuming $\beta_K(s) = 1$, again, it can be shown that there is the following recursive expression for calculating $\beta_k(s)$:

$$\beta_k(s) = \sum_{q=1}^{S} p_{sq} \, f(\mathbf{y}_{k+1} | s_{t_{k+1}} = q, \Theta) \beta_{k+1}(q) \tag{29}$$

Therefore, $\beta_k(s)$ for $k = 1, 2, ..., K$ and $s = 1, 2, ..., S$ can be easily calculated using (12) and (29).

Now, we estimate $p(s_{t_k} = s|\mathbf{Y}, \Theta)$ using the forward and backward variables. According to the conditional probability rule, $p(s_{t_k} = s|\mathbf{Y}, \Theta)$ can be expressed as

$$\begin{aligned} p(s_{t_k} = s|\mathbf{Y}, \Theta) &= \frac{p(s_{t_k} = s, \mathbf{Y}|\Theta)}{p(\mathbf{Y}|\Theta)} \\ &= \frac{p(s_{t_k} = s, \mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_K|\Theta)}{\sum_{q=1}^{S} p(s_{t_k} = q, \mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_K|\Theta)} \end{aligned} \tag{30}$$

By considering the definition of the forward and backward variables in (25) and (28), it can be seen that

$$p(s_{t_k} = s, \mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_K|\Theta) = \alpha_k(s)\beta_k(s). \tag{31}$$

Therefore, we can calculate $p(s_{t_k} = s|\mathbf{Y}, \Theta)$ for $k = 1, 2, ..., K$ and $s = 1, 2, ..., S$ as

$$p(s_{t_k} = s|\mathbf{Y}, \Theta) = \frac{\alpha_k(s)\beta_k(s)}{\sum_{q=1}^{S} \alpha_k(q)\beta_k(q)}\,. \tag{32}$$

Here, the explanation of the forward-backward procedure is finished. However, we also calculate another conditional probability which will be used in the Maximization step (*M-step*) to estimate the transition probability matrix. The probability of being in state $s$ for the $k^{th}$ signal and being in state $z$ for the $(k+1)^{th}$ signal given the observations and the model parameters is calculated as follows in a similar way to (30), (31) and (32):

$$\begin{aligned} &p(s_{t_k} = s, s_{t_{k+1}} = z|\mathbf{Y}, \Theta) = \\ &\frac{\alpha_k(s)p_{sz}f(\mathbf{y}_{k+1}|s_{t_{k+1}} = z, \Theta)\beta_{k+1}(z)}{\sum_{q=1}^{S}\sum_{r=1}^{S} \alpha_k(q)p_{qr}f(\mathbf{y}_{k+1}|s_{t_{k+1}} = r, \Theta)\beta_{k+1}(r)} \end{aligned} \tag{33}$$