A Spectral Algorithm for Inference in Hidden semi-Markov Models

Igor Melnyk **Arindam Banerjee**

Department of Computer Science and Engineering, University of Minnesota, Minneapolis, USA
{melnyk, banerjee}@cs.umn.edu

Abstract

Hidden semi-Markov models (HSMMs) are latent variable models which allow latent state persistence and can be viewed as a generalization of the popular hidden Markov models (HMMs). In this paper, we introduce a novel spectral algorithm to perform inference in HSMMs. Our approach is based on estimating certain sample moments, whose order depends only logarithmically on the maximum length of the hidden state persistence. Moreover, the algorithm requires only a few spectral decompositions and is therefore computationally efficient. Empirical evaluations on synthetic and real data demonstrate the promise of the algorithm.

1 Introduction

Hidden semi-Markov models (HSMMs) are discrete latent variable models, which allow temporal persistence of latent states and can be viewed as a generalization of the popular hidden Markov models (HMMs) [6, 15, 22]. In HSMMs, the stochastic model for the unobservable process is defined by a semi-Markov chain: latent state at the next time step is determined by the current latent state as well as time elapsed since the entry into the current state. Ability to flexibly model such latent state persistence turns out to be useful in many application areas, including anomaly detection [19, 21], activity recognition [20], and speech synthesis [24].

Given a set of training sequences, one can formulate two distinct but related problems: *learning*, i.e., estimating model parameters and *inference*, i.e., computing the probability of an observed and/or latent variable sequence. The methods proposed for learning HSMM usually follow the initial idea due to Rabiner [18] based on the modifications of the Baum-Welch algorithm [5], which are all variants of the expectation maximization (EM) framework, presented in [7]. Once the parameters are estimated, we can then perform inference using, e.g., the forward-backward algorithm of Yu et al. [23]. However, since EM, in general, has no guarantees in estimating the parameters correctly and can suffer from slow convergence, such methods can be inefficient and/or inconsistent.

In recent years, there has been an increased interest in spectral algorithms, which provide computationally efficient, local-minimum-free, provably consistent algorithms for parameter estimation and/or inference. For example, Anandkumar et al. [2, 3] have proposed spectral methods for learning the parameters of a wide class of tree-structured latent graphical models, including Gaussian mixture models, topic models, and latent Dirichlet allocation. Hsu et al. [8] have proposed an efficient spectral algorithm for inference in HMMs. The algorithm learns a so called observable representation and uses it to do inference on observable variables. The approach, however, was specific to HMMs and not easily extendable to other latent variable graphical models. Parikh et al. [17] have introduced a spectral algorithm to perform inference in latent tree graphical models with arbitrary topology, and later in [16] a general spectral inference framework for latent junction trees.

In this paper, we utilize the framework of [16] and introduce a novel spectral algorithm for inference in HSMMs. Since we address a more specific problem than [16], our results shed more light into the details of the spectral framework for HSMMs, allow for a sharper analysis, and yield a significantly more efficient algorithm than the general framework in [16]. There are two main technical contributions in this work. First, by exploiting the *homogeneity* of HSMMs we make our algorithm more efficient and accurate than if we directly follow the recipe in [16] for general graphs. In

Appearing in Proceedings of the 18^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2015, San Diego, CA, USA. JMLR: W&CP volume 38.

particular, our approach ensures that the number of matrix multiplications and inverses is fixed and independent of sequence length. Second, we show that the order of tensors in estimated observable representation depends only *logarithmically* on the maximum length of latent state persistence. In experiments, comparing our method with EM on both synthetic and real datasets, two observations stand out: first, the spectral method gets similar or better performance than EM as the number of samples increases, and the spectral method is orders of magnitude faster than EM for the datasets we consider.

Few remarks are in order about the proposed algorithm. Note that our method does not estimate model parameters explicitly but rather learns alternative representation to perform inference on observable variables. Moreover, our formulation cannot be directly used to infer hidden states, although methods such as in [14] can be potentially utilized to recover original HSMM parameters from the learned representation.

The rest of the paper is organized as follows: We introduce notation in Section 2. The inference problem and the proposed algorithm are presented in Sections 3 and 4. In Section 5, we discuss the analysis of the algorithm, followed by evaluations in Section 6 and conclusion in Section 7. Most of the technical analysis, proofs, and additional details can be found in [12].

2 Notation and Preliminaries

In this section, we cover the basic facts about the tensor algebra; more details can be found in [12], while a detailed tutorial on tensors is in [9] or [10]. A tensor is defined as a multidimensional array of data, denoted by boldface script letters, e.g., $\underset{m_1,\ldots,m_N}{\mathcal{X}} \in \mathbb{R}^{I_{m_1}\times\cdots\times I_{m_N}}$, which is Nth order tensor of dimensions $I_{m_1}\times\cdots\times I_{m_N}$. A specific dimension (or mode) is denoted by the subscript variable m_i, whose size is I_{m_i}.

Any tensor can be matrisized (or flattened) into a matrix. If we split the modes into two disjoint sets, one corresponding to rows and the other to columns, e.g., $\{m_1,\ldots,m_N\} = \{p_1,\ldots,p_K\}\cup\{q_1,\ldots,q_L\}$, then a matrisization of $\mathcal{X}$ is $\underset{p_1,\ldots,p_K q_1,\ldots,q_L}{\mathbf{X}} \in \mathbb{R}^{I_{p_1}\cdots I_{p_K}\times I_{q_1}\cdots I_{q_L}}$. Multiplication of tensors is performed along specific modes. For this, we flatten the tensor to a matrix with appropriate choice for rows and columns, perform matrix multiplication and transform the result back to tensor. The multiplication is denoted by a symbol $\times$ with an optional subscript representing the modes along which the operation is performed, e.g., multiplication along $q_1,\ldots,q_L$:

$$\underset{p_1,\ldots,p_K,r_1,\ldots,r_M}{\mathcal{Z}} = \underset{p_1,\ldots,p_K,q_1,\ldots,q_L}{\mathcal{X}} \times_{q_1,\ldots,q_L} \underset{q_1,\ldots,q_L,r_1,\ldots,r_M}{\mathcal{Y}},$$

where $\mathcal{Y}\in\mathbb{R}^{I_{q_1}\times\cdots\times I_{q_L}\times I_{r_1}\times\cdots\times I_{r_M}}$ and the resulting tensor is $\mathcal{Z}\in\mathbb{R}^{I_{p_1}\times\cdots\times I_{p_K}\times I_{r_1}\times\cdots\times I_{r_M}}$.

An important fact about tensor multiplication is that in a series of tensor multiplications the order is irrelevant as long as the multiplication is done along the matching modes: $\underset{sp}{\mathcal{X}}\times_s\left(\underset{tr}{\mathcal{Y}}\times_r\underset{rs}{\mathcal{Z}}\right)=\left(\underset{sp}{\mathcal{X}}\times_s\underset{rs}{\mathcal{Z}}\right)\times_r\underset{tr}{\mathcal{Y}}$.

Finally, we discuss the operation of tensor inversion. Tensor inverse $\mathcal{X}^{-1}$ is always defined with respect to a certain subset of the modes:

$$\underset{p_1,\ldots,p_K,q_1,\ldots,q_L}{\mathcal{X}} \times_{q_1,\ldots,q_L} \underset{p_1,\ldots,p_K,q_1,\ldots,q_L}{\mathcal{X}^{-1}} = \underset{p_1,\ldots,p_K,p_1,\ldots,p_K}{\mathcal{I}},$$

where the inversion is performed along the modes $q_1,\ldots,q_L$. Tensor on the right hand side can also be written as $\underset{p_1,\ldots,p_K}{\mathcal{I}}$ by dropping the duplicated modes. To perform tensor inversion, we first matrisize it. If the modes to be inverted along are associated with columns of the matrix, we compute the right matrix inverse, so that these modes get eliminated after the product. Otherwise, if those modes associated with rows, we compute left matrix inverse. For example, in the above equation the matrisized tensor might be of the form $\mathbf{X}\in\mathbb{R}^{I_{p_1}\cdots I_{p_K}\times I_{q_1}\cdots I_{q_L}}$, and we would compute the right matrix inverse so that the modes $q_1,\ldots,q_L$ are eliminated. If $\mathbf{X}$ has full row rank, then we compute its inverse, otherwise the pseudo-inverse. Tensorizing matrix $\mathbf{X}^{-1}$ gives us desired tensor inverse.

3 Problem Formulation: Inference in HSMMs

In this paper, we consider the problem of inference in HSMM. From a graphical model perspective, HSMM has three sets of variables: the observations $o_t\in\{1,\ldots,n_o\}$, the latent states $x_t\in\{1,\ldots,n_x\}$, and another latent variable $d_t\in\{1,\ldots,n_d\}$ which determines the length of state persistence. HSMM is specified by three conditional probability tables (CPTs): the observation/emission probability $p(o_t|x_t)$ and the state transition and the duration probabilities given by:

$$p(d_t|x_t,d_{t-1}) = \begin{cases} p(d_t|x_t) & \text{if } d_{t-1}=1 \\ \delta(d_t,d_{t-1}-1) & \text{if } d_{t-1}>1 \end{cases} \quad (1)$$

$$p(x_t|x_{t-1},d_{t-1}) = \begin{cases} p(x_t|x_{t-1}) & \text{if } d_{t-1}=1 \\ \delta(x_t,x_{t-1}) & \text{if } d_{t-1}>1 \end{cases}, \quad (2)$$

where $\delta(a,b)$ denotes the Dirac delta function: $\delta(a,b)=1$ if $a=b$ and 0 otherwise. In addition, one can consider suitable prior probabilities $p(x_0)$ and $p(d_0)$. In essence, d_t works as a down counter for state persistence. When $d_{t-1}>1$, the model remains in the same state $x_t=x_{t-1}$, while when $d_{t-1}=1$, one samples a new state x_t and the new duration in that state

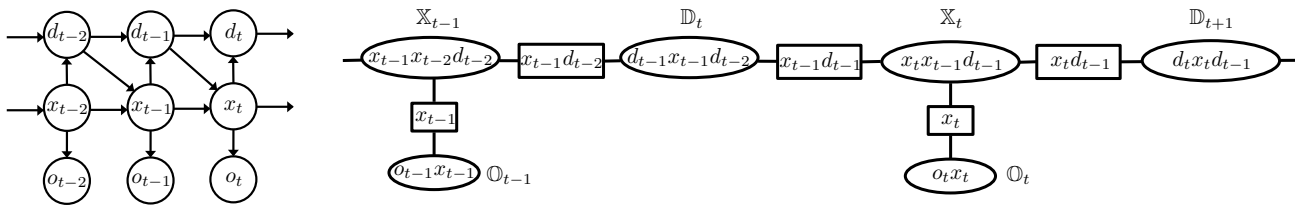

Figure 1: Left: Hidden semi-Markov Model (HSMM). Right: Junction Tree for HSMM. Ovals represent cliques, rectangles denote separators. Symbols within shapes show variables on which the corresponding objects depend.

$d_t|x_t$. For our analysis, we assume $p(d_t|x_t, d_{t-1}=1)$ to be a multinomial distribution over $\{1,\dots,n_d\}$ where n_d denotes the largest duration of state persistence.

The considered inference problem can be posed as follows: given a set of sequences $\{\mathbf{S}^1,\dots,\mathbf{S}^N\}$ drawn independently from the HSMM model, where each sequence is $\mathbf{S}^i=\{o_1^i,\dots,o_{T_i}^i\}, i=1,\dots,N$, our goal is to develop a provably correct spectral algorithm for computing $p(\mathbf{S}^{test})$ of any given test sequence $\mathbf{S}^{test}=(o_1^{test},\dots,o_T^{test})$.

We start by considering the matrix forms of the HSMM parameters and writing the computations in tensor notation, as introduced in Section 2. Specifically, $p(d_t|x_t,d_{t-1}=1)$ is denoted as $D\in\mathbb{R}^{n_d\times n_x}$, $p(x_t|x_{t-1},d_{t-1}=1)$ is denoted as $\mathcal{X}\in\mathbb{R}^{n_x\times n_x}$, and $p(o_t|x_t)$ as $O\in\mathbb{R}^{n_o\times n_x}$. We make the following assumptions on the HSMM parameters:

Assumptions 3.1

1. *$\mathcal{X}$ is full rank and has non-zero probability of visiting any state from any other state.*
2. *D has a non-zero probability of any duration in any state.*
3. *O is full column rank and $n_x\le n_o$.*

To express the joint probability $p(o_1,\dots,o_T)$ for an observation sequence in tensor form, we utilize the junction tree [4] corresponding to the graphical model of HSMM (see Figure 1). We proceed by embedding the clique CPTs of the junction tree into tensors. For example, the clique $\mathbb{X}_t$, containing the CPT of $p(x_t|x_{t-1},d_{t-1})$ is embedded into tensor $\underset{x_t|x_{t-1}d_{t-1}}{\mathcal{X}}$. For ease of exposition, the tensor's modes are named based on the variables on which the tensor depends. Similarly, we embed the clique $\mathbb{D}_t$ with its CPT $p(d_t|x_t,d_{t-1})$ into tensor $\underset{d_t|x_td_{t-1}}{\mathcal{D}}$, and $\mathbb{O}_t$ containing $p(o_t|x_t)$ into tensor $\underset{o_t|x_t}{\mathcal{O}}$.

If we denote the joint probability of the observed sequence $p(o_1,\dots,o_T)$ as $\underset{o_1,\dots,o_T}{\mathcal{P}}$ then the message passing algorithm for the junction tree in Figure 1 can be represented as tensor multiplications:

$$\underset{o_1,\dots,o_T}{\mathcal{P}} = \prod_t \underset{d_{t-1}|x_{t-1}x_{t-1}d_{t-2}}{\mathcal{D}} \times_{x_{t-1}d_{t-1}} \times_{x_{t-1}d_{t-1}} \left(\underset{x_tx_t|x_{t-1}d_{t-1}d_{t-1}}{\mathcal{X}} \times_{x_t} \underset{o_t|x_t}{\mathcal{O}} \right), \quad (3)$$

where, for simplicity, we denoted by $\prod_t$ the tensor product over multiple time steps. Observe that the tensors are multiplied along the modes (dimensions) which are the separator variables between the cliques in Figure 1. A certain mode of the tensor is duplicated the number of times such variable appears in the separators adjacent to the clique, ensuring that tensor multiplication remains valid. In what follows, we represent expression (3) in the observable form so that all the factors can be estimated directly from data using certain sample moments and provide a practical algorithm implementing these ideas.

4 Spectral Algorithm for Inference in HSMM

Observe that the expression for the joint probability in (3) depends on the unknown model parameters. Our goal is to change the tensor representation such that $\underset{o_1,\dots,o_T}{\mathcal{P}}$ can be written in terms of the quantities directly computable from data. To that end, we follow [16] and between every two neighboring factors in (3) introduce an identity tensor with the modes corresponding to the modes along which the multiplication is performed. Intuitively, this operation corresponds to the marginalization step, expressed in tensor form. For example, consider a part of expression (3) after introducing identity tensors:

$$\times \underset{x_{t-1}d_{t-2}}{\mathcal{I}} \times_{x_{t-1}d_{t-2}} \underset{d_{t-1}|x_{t-1}x_{t-1}d_{t-2}}{\mathcal{D}} \times_{x_{t-1}d_{t-1}} \underset{x_{t-1}d_{t-1}}{\mathcal{I}} \times_{x_{t-1}d_{t-1}}$$
$$\times_{x_{t-1}d_{t-1}} \left(\underset{x_tx_t|x_{t-1}d_{t-1}d_{t-1}}{\mathcal{X}} \times_{x_t} \underset{x_t}{\mathcal{I}} \times_{x_t} \underset{o_tx_t}{\mathcal{O}} \right) \times_{x_td_{t-1}} \underset{x_td_{t-1}}{\mathcal{I}} \times$$

where all the identity tensors have duplicated modes but are not shown. Now rewrite each of the identity tensors as a multiplication of some factor times its inverse. For example, $\underset{x_t}{\mathcal{I}} = \underset{\omega_{x_t}x_t}{\mathcal{F}} \times_{\omega_{x_t}} \underset{\omega_{x_t}x_t}{\mathcal{F}^{-1}}$, for some invertible factor $\underset{\omega_{x_t}x_t}{\mathcal{F}}$. Note that the choice of mode x_t

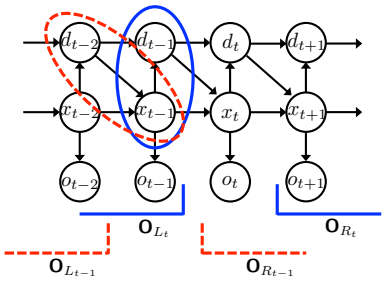

Figure 2: Conditional independence in HSMM. $\mathbf{O}_{L_t}$ and $\mathbf{O}_{R_t}$ are independent conditioned on $x_{t-1}d_{t-1}$, similarly, $\mathbf{O}_{L_{t-1}}$ and $\mathbf{O}_{R_{t-1}}$ are conditionally independent given $x_{t-1}d_{t-2}$. We defined $\mathbf{O}_{L_t} = \{\ldots, o_{t-2}, o_{t-1}\}$ and $\mathbf{O}_{R_t} = \{o_{t+1}, o_{t+2}, \ldots\}$.

is fixed and is determined by the modes of the identity tensor $\underset{x_t}{\mathcal{I}}$, while the mode ω_{x_t} is not fixed and we have a freedom in selecting it. Moreover, since the tensor inversion is done along the mode ω_{x_t}, if the matrix $\mathbf{F}$ has its rows associated with ω_{x_t}, then this matrix must have full column rank for the inverse to exist and for the product $\mathbf{F}^{-1}\mathbf{F}$ to equal identity matrix (see Section 2). Based on this, we choose the modes ω_{x_t} such that (i) ω_{x_t} are the observed variables, (ii) $\underset{\omega_{x_t} x_t}{\mathcal{F}}$ is invertible and (iii) interpret this factor as corresponding to a conditional probability distribution, i.e., $p(\omega_{x_t}|x_t)$ and write as $\underset{\omega_{x_t}|x_t}{\mathcal{F}}$.

After expanding each of the identity tensors, regrouping the factors and recalling that in a series of tensor multiplication the order is irrelevant, we can identify the tensors

$$\underset{\omega_{x_{t-1}d_{t-2}}\omega_{x_{t-1}d_{t-1}}}{\tilde{\mathcal{D}}} = \underset{\omega_{x_{t-1}d_{t-2}}|x_{t-1}d_{t-2}}{\mathcal{F}^{-1}} \times_{x_{t-1}d_{t-2}} \underset{d_{t-1}|x_{t-1}x_{t-1}d_{t-2}}{\mathcal{D}} \times_{x_{t-1}d_{t-1}} \underset{\omega_{x_{t-1}d_{t-1}}|x_{t-1}d_{t-1}}{\mathcal{F}},$$

$$\underset{\omega_{x_{t-1}d_{t-1}}\omega_{x_t}\omega_{x_t}d_{t-1}}{\tilde{\mathcal{X}}} = \underset{\omega_{x_{t-1}d_{t-1}}|x_{t-1}d_{t-1}}{\mathcal{F}^{-1}} \times_{x_{t-1}d_{t-1}} \left(\underset{x_t x_t|x_{t-1}d_{t-1}d_{t-1}}{\mathcal{X}} \times_{x_t} \underset{\omega_{x_t}|x_t}{\mathcal{F}}\right) \times_{x_t d_{t-1}} \underset{\omega_{x_t d_{t-1}}|x_t d_{t-1}}{\mathcal{F}}$$

and $\underset{\omega_{x_t} o_t}{\tilde{\mathcal{O}}} = \underset{\omega_{x_t}|x_t}{\mathcal{F}^{-1}} \times_{x_t} \underset{o_t|x_t}{\mathcal{O}}$. Note that although each of the above tensors depends only on the observed variables ω, it is not clear yet how to estimate them: the expressions on the right depend on the unknown model parameters, while the tensors on the left do not correspond to valid probability distributions (due to the presence of inverses $\mathcal{F}^{-1}$). For example, $\underset{\omega_{x_{t-1}d_{t-2}}\omega_{x_{t-1}d_{t-1}}}{\tilde{\mathcal{D}}}$ is not a tensor form of $p(\omega_{x_{t-1}d_{t-2}}, \omega_{x_{t-1}d_{t-1}})$.

Next, we discuss the choice of the observable set ω in the factors $\mathcal{F}$. From Figure 1 we can see that there are three types of separators: $x_{t-1}d_{t-1}$, $x_t d_{t-1}$ and x_t, consequently, there are three types of identity tensors which we introduced: $\underset{x_{t-1}d_{t-1}}{\mathcal{I}}$, $\underset{x_t d_{t-1}}{\mathcal{I}}$ and $\underset{x_t}{\mathcal{I}}$. Therefore, we need to define three types of observable sets $\omega_{x_{t-1}d_{t-1}}$, $\omega_{x_t d_{t-1}}$ and ω_{x_t}. There could be multiple choices for these sets, one of them is $\omega_{x_{t-1}d_{t-1}} = \omega_{x_t d_{t-1}} = \{o_{t+1}, o_{t+2}, \ldots\}$ and $\omega_{x_t} = o_t$ for all t. The detailed description of how and what number of these observations to select is deferred until Section 5. In what follows, we define $\mathbf{O}_{R_t} := \{o_{t+1}, o_{t+2}, \ldots\}$, to emphasize that this is a set of observations starting at time stamp $t+1$ and going to the right (or forward in time), see Figure 2. With these definitions, we can now rewrite (3) in the form:

$$\underset{o_1,\ldots,o_T}{\mathcal{P}} = \prod_t \underset{\mathbf{O}_{R_{t-1}}\mathbf{O}_{R_t}}{\tilde{\mathcal{D}}} \times_{\mathbf{O}_{R_t}} \left(\underset{\mathbf{O}_{R_t} o_t \mathbf{O}_{R_t}}{\tilde{\mathcal{X}}} \times_{o_t} \underset{o_t o_t}{\tilde{\mathcal{O}}}\right). \quad (4)$$

Comparing (3) and (4) we see that the above equation expresses the joint probability distribution in the observable form. As noted above, we cannot yet use this formula in practice since we do not know how to compute the transformed tensors. In what follows, we show how to estimate such tensors directly from data, without the need of the model parameters.

4.1 Estimation of Observable Tensors

Consider the tensor $\tilde{\mathcal{D}}$:

$$\underset{\mathbf{O}_{R_{t-1}}\mathbf{O}_{R_t}}{\tilde{\mathcal{D}}} = \underset{\mathbf{O}_{R_{t-1}}|x_{t-1}d_{t-2}}{\mathcal{F}^{-1}} \times_{x_{t-1}d_{t-2}} \underset{d_{t-1}|x_{t-1}x_{t-1}d_{t-2}}{\mathcal{D}} \times_{x_{t-1}d_{t-1}} \underset{\mathbf{O}_{R_t}|x_{t-1}d_{t-1}}{\mathcal{F}}, \quad (5)$$

whose modes are the observable variables $\mathbf{O}_{R_{t-1}}$ and $\mathbf{O}_{R_t}$. To estimate this tensor from data, consider $\mathbf{O}_{L_{t-1}}$, a set of the observed variables such that $\mathbf{O}_{L_{t-1}}$ and $\mathbf{O}_{R_{t-1}}$ are independent, conditioned on $x_{t-1}d_{t-2}$ (see Figure 2). The tensor form of this relationship is:

$$\underset{\mathbf{O}_{L_{t-1}}\mathbf{O}_{R_{t-1}}}{\mathcal{M}} = \underset{\mathbf{O}_{L_{t-1}}|x_{t-1}d_{t-2}}{\mathcal{F}} \times_{x_{t-1}d_{t-2}} \underset{\mathbf{O}_{R_{t-1}}|x_{t-1}d_{t-2}}{\mathcal{F}} \times_{x_{t-1}d_{t-2}} \underset{x_{t-1}d_{t-2}}{\mathcal{K}}, \quad (6)$$

where tensor $\mathcal{K}$ represents the marginal $p(x_{t-1}, d_{t-2})$. Note that, though not shown, the modes x_{t-1} and d_{t-2} need to appear twice in $\mathcal{K}$, since it interacts with both other terms. The set $\mathbf{O}_{L_{t-1}}$ is defined in a way similar to $\mathbf{O}_{R_t}$ but with the set of observations starting at time stamp $t-2$ and going to the left (or backward in time), i.e., $\mathbf{O}_{L_{t-1}} := \{\ldots, o_{t-3}, o_{t-2}\}$ (see Figure 2).

Next, express the inverse of the tensor $\underset{\mathbf{O}_{R_{t-1}}|x_{t-1}d_{t-2}}{\mathcal{F}}$

from (6) and substitute back to (5).

$$\begin{aligned}\underset{\mathbf{o}_{R_{t-1}}\mathbf{o}_{R_t}}{\tilde{\mathcal{D}}} &= \underset{\mathbf{o}_{L_{t-1}}\mathbf{o}_{R_{t-1}}}{\mathcal{M}^{-1}} \times_{\mathbf{o}_{L_{t-1}}} \underset{\mathbf{o}_{L_{t-1}}|x_{t-1}d_{t-2}}{\mathcal{F}} \times_{x_{t-1}d_{t-2}} \\ &\quad \underset{x_{t-1}d_{t-2}}{\mathcal{K}} \times_{x_{t-1}d_{t-2}} \underset{d_{t-1}|x_{t-1}x_{t-1}d_{t-2}}{\mathcal{D}} \times_{x_{t-1}d_{t-1}} \underset{\mathbf{o}_{R_t}|x_{t-1}d_{t-1}}{\mathcal{F}} \\ &= \underset{\mathbf{o}_{L_{t-1}}\mathbf{o}_{R_{t-1}}}{\mathcal{M}^{-1}} \times_{\mathbf{o}_{L_{t-1}}} \underset{\mathbf{o}_{L_{t-1}}\mathbf{o}_{R_t}}{\mathcal{M}}, \end{aligned} \tag{7}$$

where we eliminated all the latent variables by multiplying the last four terms on the first line. Observe that the tensors $\underset{\mathbf{o}_{L_{t-1}}\mathbf{o}_{R_{t-1}}}{\mathcal{M}}$ and $\underset{\mathbf{o}_{L_{t-1}}\mathbf{o}_{R_t}}{\mathcal{M}}$ represent valid probability distributions and though they are defined using unknown model parameters, we can readily estimate them from data. For example, $\underset{\mathbf{o}_{L_{t-1}}\mathbf{o}_{R_t}}{\mathcal{M}}$ is a tensor, where each entry is computed from the frequency of co-occurrence of tuples of the observed symbols $\{\ldots, o_{t-3}, o_{t-2}, o_{t+1}, o_{t+2}, \ldots\}$. The specific number and order of this symbols will be determined in Section 5. Similar derivations can be used to find the estimates for tensors $\underset{\mathbf{o}_{R_t}o_t\mathbf{o}_{R_t}}{\tilde{\mathcal{X}}} = \underset{\mathbf{o}_{L_t}\mathbf{o}_{R_t}}{\mathcal{M}^{-1}} \times_{\mathbf{o}_{L_t}} \underset{\mathbf{o}_{L_t}\mathbf{o}_{R_t}o_t}{\mathcal{M}}$ and $\underset{o_to_t}{\tilde{\mathcal{O}}} = \underset{o_to_{t+1}}{\mathcal{M}^{-1}} \times_{o_{t+1}} \underset{o_to_{t+1}}{\mathcal{M}}$ (see [12] for details).

4.2 Spectral Algorithm

In the previous section, we expressed the tensors $\mathcal{D}$, $\mathcal{X}$ and $\mathcal{O}$ in terms of the moments directly computable from data, so now we can obtain the spectral algorithm to compute $\underset{o_1,\ldots,o_T}{\mathcal{P}}$ entirely using the observed variables. The basic version of the spectral HSMM algorithm, which follows from the framework of [16], can be described as a two step process: in the learning step, compute $\underset{\mathbf{o}_{R_{t-1}}\mathbf{o}_{R_t}}{\tilde{\mathcal{D}}}$, $\underset{\mathbf{o}_{R_{t-1}}o_t\mathbf{o}_{R_t}}{\tilde{\mathcal{X}}}$ and $\underset{o_to_t}{\tilde{\mathcal{O}}}$ for all t using the data. In the inference step, use (4) to compute $p(\mathbf{S}^{test})$.

If we denote the number of required observations in $\mathbf{O}_R$ or $\mathbf{O}_L$ as ℓ (in Section 5 we will show that $\ell = \lceil 1 + \frac{\log n_d}{\log n_x} \rceil$), then the algorithm's computational complexity is $\mathcal{O}\left((n_o^{3\ell} + N\ell)T\right)$ for learning and $\mathcal{O}\left(n_o^{3\ell}T\right)$ for inference, mainly determined by tensor inversions, multiplications, and the estimation of tensors $\mathcal{M}$ in (7) and in $\underset{\mathbf{o}_{R_t}o_t\mathbf{o}_{R_t}}{\tilde{\mathcal{X}}}$ and $\underset{o_to_t}{\tilde{\mathcal{O}}}$ for all t. Here, N is the number of training samples and T is the length of the observation sequences. Note that for large ℓ accurate estimation of tensors $\mathcal{M}$ for each t will require large number of training sequences which might not be available, leading to inaccurate and unstable computations.

A novel aspect of our work is the improvement of the accuracy and efficiency of the basic algorithm mentioned above by exploiting the homogeneity property of HSMM and estimating the tensors $\tilde{\mathcal{X}}$, $\tilde{\mathcal{D}}$ and $\tilde{\mathcal{O}}$ in the batch, by averaging across all t. Thus, we compute only three tensors for all t, as opposed to computing these tensors for each t. For example, using (7), the batch form of tensor $\tilde{\mathcal{D}}$ takes a form:

$$\tilde{\mathcal{D}} = \left(\frac{1}{T}\sum_t \underset{\mathbf{o}_{L_{t-1}}\mathbf{o}_{R_{t-1}}}{\mathcal{M}}\right)^{-1} \times_{\mathbf{o}_L} \left(\frac{1}{T}\sum_t \underset{\mathbf{o}_{L_{t-1}}\mathbf{o}_{R_t}}{\mathcal{M}}\right),$$

where $\mathbf{o}_L$ denotes a generic mode of the averaged tensor $\mathcal{M}$, corresponding to $\mathbf{o}_{L_{t-1}}$ for all t. Similar expressions can be defined for other tensors; the detailed derivations are provided in [12]. This modification reduced the computational complexity of the learning phase to $\mathcal{O}\left((n_o^{2\ell} + N\ell)T\right)$ (the cost of inference step still remains $\mathcal{O}\left(n_o^{3\ell}T\right)$), where the main operations are now tensor additions and estimation of tensors $\mathcal{M}$. Note that the number of inverses and multiplications in the learning phase is now fixed and independent of sequence length. This is in contrast to the basic version of algorithm, which involves tensor inverses and multiplications at every step t (e.g., see (7)). Moreover, such averaging improves the accuracy of the resulting algorithm since the estimates obtained in this form have lower variance, which in turn ensures that the computed inverses are more stable and accurate.

5 Rank Analysis of Observable Tensors

In Section 4.1, when we derived $\underset{\mathbf{o}_{R_{t-1}}\mathbf{o}_{R_t}}{\tilde{\mathcal{D}}}$, $\underset{\mathbf{o}_{R_{t-1}}o_t\mathbf{o}_{R_t}}{\tilde{\mathcal{X}}}$ and $\underset{o_to_t}{\tilde{\mathcal{O}}}$, we glossed over the question of the existence of tensor inverses $\underset{\mathbf{o}_{L_{t-1}}\mathbf{o}_{R_{t-1}}}{\mathcal{M}^{-1}}$, $\underset{\mathbf{o}_{L_t}\mathbf{o}_{R_t}}{\mathcal{M}^{-1}}$ and $\underset{o_to_{t+1}}{\mathcal{M}^{-1}}$. In this section, our task is to analyze the rank structure of these tensors and impose restrictions on the sets $\mathbf{O}_L$ and $\mathbf{O}_R$ to ensure that the rank conditions are satisfied. For example, consider equation (7) and expand all its terms using (6) to get

$$\begin{aligned}\underset{\mathbf{o}_{R_{t-1}}\mathbf{o}_{R_t}}{\tilde{\mathcal{D}}} &= \underset{\mathbf{o}_{R_{t-1}}|x_{t-1}d_{t-2}}{\mathcal{F}^{-1}} \times \underset{\mathbf{o}_{L_{t-1}}|x_{t-1}d_{t-2}}{\mathcal{F}^{-1}} \times \underset{x_{t-1}d_{t-2}}{\mathcal{K}^{-1}} \times \\ &\quad \times \underset{x_{t-1}d_{t-2}}{\mathcal{K}} \times \underset{\mathbf{o}_{L_{t-1}}|x_{t-1}d_{t-2}}{\mathcal{F}} \times \underset{d_{t-1}|x_{t-1}d_{t-2}}{\mathcal{D}} \times \underset{\mathbf{o}_{R_t}|x_{t-1}d_{t-1}}{\mathcal{F}},\end{aligned}$$

where we dropped the multiplication subscripts and the duplicated modes, which can be inferred from the context. Observe, that in order for the above equation to produce (5), the terms in the middle must multiply out into identity tensor, i.e.,

$$\underset{x_{t-1}d_{t-2}}{\mathcal{I}} = \underset{x_{t-1}d_{t-2}}{\mathcal{K}^{-1}} \times_{x_{t-1}d_{t-2}} \underset{x_{t-1}d_{t-2}}{\mathcal{K}} \tag{8}$$

$$\underset{x_{t-1}d_{t-2}}{\mathcal{I}} = \underset{\mathbf{o}_{L_{t-1}}|x_{t-1}d_{t-2}}{\mathcal{F}^{-1}} \times_{\mathbf{o}_{L_{t-1}}} \underset{\mathbf{o}_{L_{t-1}}|x_{t-1}d_{t-2}}{\mathcal{F}}. \tag{9}$$

Moreover, recall that $\underset{\mathbf{o}_{R_{t-1}}|x_{t-1}d_{t-2}}{\mathcal{F}}$ was originally introduced as part of the identity tensor

$$\underset{x_{t-1}d_{t-2}}{\mathcal{I}} = \underset{\mathbf{o}_{R_{t-1}}|x_{t-1}d_{t-2}}{\mathcal{F}^{-1}} \times_{\mathbf{o}_{R_{t-1}}} \underset{\mathbf{o}_{R_{t-1}}|x_{t-1}d_{t-2}}{\mathcal{F}}, \tag{10}$$

therefore, we can conclude that for (7) to exist, the identity statements in (8), (9) and (10) must be satisfied. These statements have implications for the ranks of $\underset{x_{t-1}d_{t-2}}{\mathcal{K}}$, $\underset{\mathbf{O}_{L_{t-1}}|x_{t-1}d_{t-2}}{\mathcal{F}}$ and $\underset{\mathbf{O}_{R_{t-1}}|x_{t-1}d_{t-2}}{\mathcal{F}}$, which in turn determine the length of the observation sequences $\mathbf{O}_{L_{t-1}}$ and $\mathbf{O}_{R_{t-1}}$.

Since $\underset{x_{t-1}d_{t-2}}{\mathcal{K}}$ represents a distribution $p(x_{t-1}d_{t-2})$, its matrisized version is a diagonal matrix with $p(x_{t-1}d_{t-2})$ on the diagonal. Using statements 1 and 2 in Assumptions 3.1, it can be concluded that the diagonal elements in this matrix are non-zero and it has rank $n_x n_d$, it is thus invertible and so (8) is satisfied.

Next, consider (9) and recall from Section 2 that if we matrisize the tensor as $\underset{\mathbf{O}_{L_{t-1}}|x_{t-1}d_{t-2}}{\mathbf{F}} \in \mathbb{R}^{n_o^{|\mathbf{O}_{L_{t-1}}|} \times n_x n_d}$ then $\mathbf{F}$ must have full column rank $n_x n_d$ for the proper inverse to exist, implying $n_o^{|\mathbf{O}_{L_{t-1}}|} \geq n_x n_d$. Similarly, $\underset{\mathbf{O}_{R_{t-1}}|x_{t-1}d_{t-2}}{\mathcal{F}}$ in (10) must have rank $n_x n_d$. As a consequence, the tensor $\underset{\mathbf{O}_{L_{t-1}}\mathbf{O}_{R_{t-1}}}{\mathcal{M}}$ will have rank $n_x n_d$ and, in general, is rank-deficient. The argument above can also be used to show that $\underset{\mathbf{O}_{L_t}\mathbf{O}_{R_t}}{\mathcal{M}}$ has rank $n_x n_d$ and the rank of $\underset{o_t o_{t+1}}{\mathcal{M}}$ is n_x (see [12] for mode details).

The key unknowns now are the sets of the observed variables $\mathbf{O}_R$ and $\mathbf{O}_L$ that must be appropriately selected for the corresponding tensors to have rank $n_x n_d$. In the following, we discuss the results for the tensor $\underset{\mathbf{O}_{R_{t-1}}|x-1d-2}{\mathcal{F}}$, however HSMM homogeneity property allows to transfer the result for any t. Moreover, the analysis for tensors with $\mathbf{O}_L$ is similar and omitted. Recall that we defined $\mathbf{O}_{R_{t-1}} = \{o_t, o_{t+1}, \ldots\}$. As the main contribution of our work, we established that if we select the observations o_t non-sequentially with gaps that grow exponentially with the state size n_x then the following result holds for all t:

Theorem 5.1 *Let the number of observations be* $|\mathbf{O}_{R_{t-1}}| = \ell$ *and define the set of indices* $\mathcal{S} = \big\{\max\big[t,\ t+(n_d-1)-(n_x^i-1)\big]\ \mid i=0,\ldots,\ell-1\big\}$, *such that* $\mathbf{O}_{R_{t-1}} = \{o_k | k \in \mathcal{S}\}$ *then the rank of tensor* $\underset{\mathbf{O}_{R_{t-1}}|x_{t-1}d_{t-2}}{\mathcal{F}}$ *is* $\min[n_x^\ell,\ n_x n_d]$.

As a consequence of this result, to achieve the rank $n_x n_d$ we will require $\ell = \lceil 1 + \frac{\log n_d}{\log n_x} \rceil$ observations, since we need to ensure $n_x^\ell = n_x n_d$. The span of the selected observations is n_d, while their number is only logarithmic in n_d. For example, consider the estimation of tensor $\underset{\mathbf{O}_{L_{t-1}}\mathbf{O}_{R_{t-1}}}{\mathcal{M}}$ for an HSMM with $n_x = 3$ and $n_d = 20$. In this case $\ell = 4$ and $\mathbf{O}_{L_{t-1}} = \{o_{t-21}, o_{t-19}, o_{t-13}, o_{t-2}\}$ and $\mathbf{O}_{R_{t-1}} = \{o_t, o_{t+11}, o_{t+17}, o_{t+19}\}$. Figure 3 illustrates this example. We note that the requirement for the span of the selected observations to be n_d, which is a maximum state persistence, is to ensure that for a given time stamp t, we select the observations far enough to the right and left of it so that those observations are likely to be sampled from different hidden states.

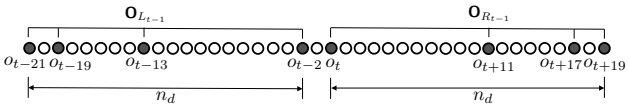

Figure 3: Observations required to estimate $\underset{\mathbf{O}_{L_{t-1}}\mathbf{O}_{R_{t-1}}}{\mathcal{M}}$ from data for HSMM with $n_x = 3$, $n_d = 20$.

5.1 Proof sketch of Theorem 5.1

In the following we present main ideas to prove Theorem 5.1, all the details can be found in [12]. Define by $\mathbf{X}_{R_{t-1}} = \{x_t, x_{t+1}, \ldots\}$, the sequence of hidden states corresponding to $\mathbf{O}_{R_{t-1}} = \{o_t, o_{t+1}, \ldots\}$. Using conditional independence property of graphical model in Figure 1, i.e., $\mathbf{O}_{R_{t-1}}$ and $x_{t-1}d_{t-2}$ are independent given $\mathbf{X}_{R_{t-1}}$, we can write:

$$\underset{\mathbf{O}_{R_{t-1}}|x_{t-1}d_{t-2}}{\mathcal{F}} = \underset{\mathbf{O}_{R_{t-1}}|\mathbf{X}_{R_{t-1}}}{\mathcal{Q}} \times \underset{\mathbf{X}_{R_{t-1}}|x_{t-1}d_{t-2}}{\mathcal{T}}, \quad (11)$$

for some tensors $\mathcal{Q}$ and $\mathcal{T}$, representing the appropriate probability distributions.

Denoting $\ell = |\mathbf{O}_{R_{t-1}}| = |\mathbf{X}_{R_{t-1}}|$, it can be verified that the matrisized form of $\mathcal{Q}$ in (11) can be written as $\mathbf{Q} = \otimes_\ell O \in \mathbb{R}^{n_o^\ell \times n_x^\ell}$, a Kronecker product of the observation matrix O with itself ℓ times. According to the Assumptions 3.1, $rank(O) = n_x$ and $n_x \leq n_o$, and using the rank property of the Kronecker product, we infer that $rank(\mathbf{Q}) = n_x^\ell$.

Combining the above conclusion with the fact that the matrisized form of the other two tensors in (11) is $\mathbf{F} \in \mathbb{R}^{n_o^\ell \times n_x n_d}$ and $\mathbf{T} \in \mathbb{R}^{n_x^\ell \times n_x n_d}$, to ensure the rank of $\mathcal{F}$ is $n_x n_d$, we need to select a set of variables $\mathbf{X}_{R_{t+1}}$ so that $rank\Big(\underset{\mathbf{X}_{R_{t-1}}|x_{t-1}d_{t-2}}{\mathbf{T}}\Big) = n_x n_d$ with the condition that $n_x^\ell \geq n_x n_d$. Thus, the problem of the analysis of the rank structure of tensor $\underset{\mathbf{O}_{R_{t-1}}|x_{t-1}d_{t-2}}{\mathcal{F}}$ was translated to the problem of rank structure of matrix $\underset{\mathbf{X}_{R_{t-1}}|x_{t-1}d_{t-2}}{\mathbf{T}}$. The main idea of the analysis of such a matrix is to understand the mechanism how the rank changes as the size of $\mathbf{X}_{R_{t-1}}$ increases.

Starting with a matrix $\underset{x_t|x_{t-1}d_{t-2}}{\mathbf{T}}$, its rank is n_x since it is a matrisized version of the transition probability of the model and statement 1 in Assumptions 3.1 guarantees it has full rank. Next, if we add a hidden state

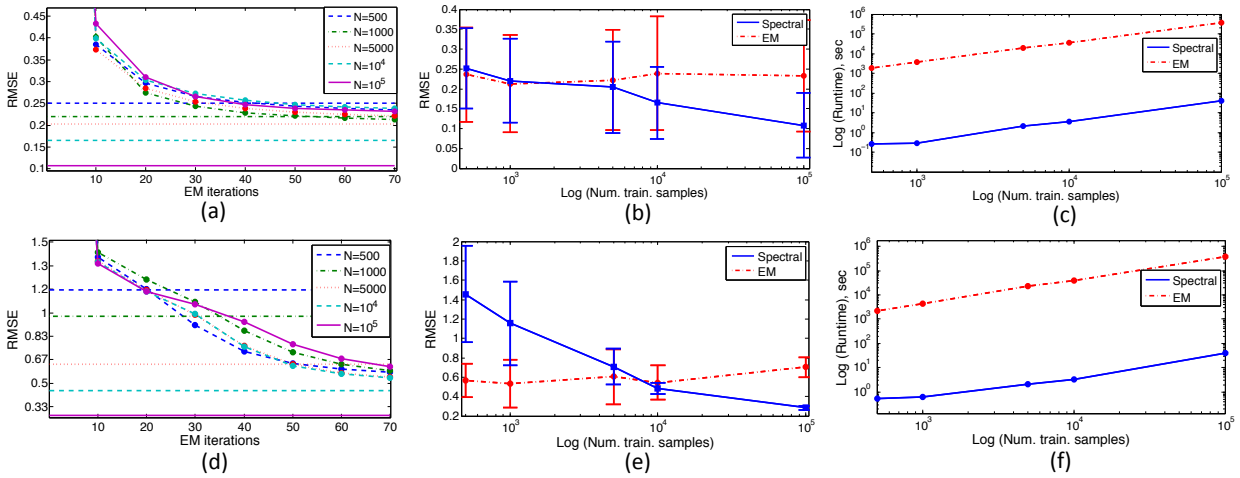

Figure 4: Performance of the spectral algorithm and EM on synthetic data generated from HSMM with $n_o = 3, n_x = 2, n_d = 2$ (top row) and $n_o = 5, n_x = 4, n_d = 6$ (bottom row). (a), (d): Error for EM across different iterations for various training datasets. The straight lines show the performance for spectral method. (b), (e): Average error and one standard deviation over 100 runs for EM after convergence and spectral algorithm across different number of training data. (c), (f): Runtime, in seconds, for both methods.

at the consecutive time stamp, i.e., $\mathbf{T}_{x_{t+1},x_t|x_{t-1}d_{t-2}} \in \mathbb{R}^{n_x^2 \times n_x n_d}$ then it turns out that the rank of this matrix is $2n_x$ (assuming that $n_x n_d \geq 2n_x$). In general, in can be shown that the matrix $\mathbf{T}$ constructed with the hidden states at consecutive time stamp allows only linear growth of its rank. Consequently, to have rank $n_x n_d$ would require $\ell = n_d$ observations.

On the other hand, if we consider $\mathbf{T}_{x_{t+\delta},x_t|x_{t-1}d_{t-2}} \in \mathbb{R}^{n_x^2 \times n_x n_d}$, for $\delta = 1, \ldots, n_x - 1$, then the rank of this matrix is $(\delta + 1)n_x$, with the largest rank of n_x^2 (if $n_x n_d \geq n_x^2$). In general, it can be proved that if we include hidden state at time stamp j-th, $j = 1, \ldots, \ell$ by skipping $n_x^j - n_x^{j-1} - 1$ time stamps, we will be increasing rank as n_x^j, i.e., exponentially fast up to a maximum achievable rank $n_x n_d$. As a result, the number of required observations is only $\ell = \lceil 1 + \frac{\log n_d}{\log n_x} \rceil$. To illustrate this, refer back to Figure 3 where $n_x = 3$ and $n_d = 20$. The $\ell = 4$ observations we include for $\mathbf{O}_{R_{t-1}}$ are o_{t+19}, o_{t+17}, o_{t+11} and o_t, which lead to the following rank growth: 3, 9, 27, 60.

6 Experiments

In this section we evaluated the performance of the proposed algorithm both on synthetic as well as real datasets and compared its performance to a standard EM algorithm.

6.1 Synthetic Data

Using synthetic data, we compared the estimation accuracy and the runtime of the spectral algorithm with EM. For this, we defined two HSMMs, one with $n_o = 3, n_x = 2, n_d = 2$ and another with $n_o = 5, n_x = 4, n_d = 6$. For each model, we generated a set of $N = \{500, 1000, 5000, 10^4, 10^5\}$ training and $N = 1000$ testing sequences, each of length $T = 100$. The accuracy of estimating likelihood for each testing sequence was measured using the relative deviation from the true likelihood, i.e., $\epsilon_i = \frac{|\hat{p}(\mathbf{S}_i^{test}) - p(\mathbf{S}_i^{test})|}{p(\mathbf{S}_i^{test})}$ for $i = 1, \ldots, 1000$. Given $N = 1000$ such values, we then computed the final score, which is the root-mean-square error (RMSE) across all the testing sequences, $\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} \epsilon_i^2}$. Figure 4 shows results.

It can be observed from plots (b) and (e) in Figure 4 that with the small training set, EM achieves smaller errors, while as the number of training samples increases, the spectral method becomes more accurate, outperforming EM. Also, comparing the plots (a), (b) with (d) and (e), we can conclude that for larger models the spectral method requires more data in order to achieve same or better accuracy than EM. This is expected since the sizes of estimated tensors grow with the model size. Moreover, the plots (c) and (d) in Figure 4 show that spectral method is several orders of magnitude faster than EM. Given the above results, we can conclude that for small datasets EM is a preferable algorithm, while for large data, the spectral algorithm is a better choice, it achieves higher accuracy and requires significantly smaller computational resources.

6.2 Real Data

We also compared the performance of the spectral algorithm and EM on real NASA flight dataset [1], containing over 180000 flights of 35 aircrafts from a de-

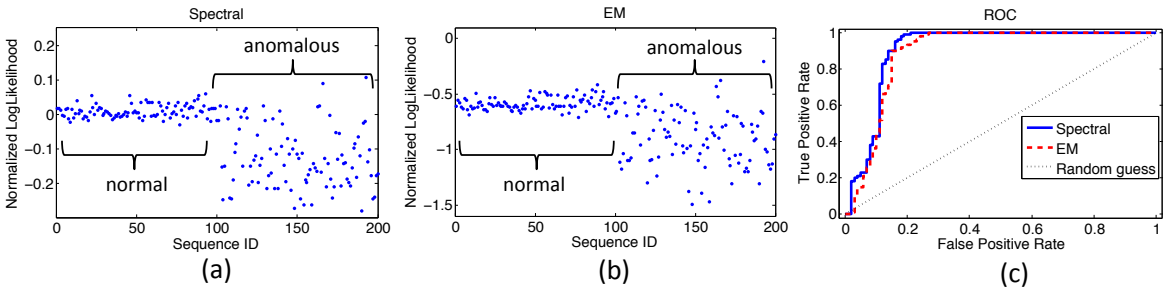

Figure 5: Evaluation of the spectral algorithm and EM on real data. (a) and (b): Normalized joint loglikelihood computed by spectral algorithm (a) and EM (b) for a set of 200 test flights, with 100 normal and 100 anomalous. HSMM parameters: $n_o = 9, n_x = 8, n_d = 40$ (c): The Receiver Operating Characteristic (ROC) curve, illustrating classification accuracy of the algorithms.

Parameters		n_o=9, n_x=8, n_d=40	n_o=9, n_x=7, n_d=30	n_o=9, n_x=6, n_d=20	n_o=9, n_x=5, n_d=10
Time	Spectral	6.8 hours	6.4 hours	6.4 hours	6.3 hours
	EM	$>$ 2 days	$>$ 2 days	$>$ 2 days	$>$ 2 days
AUC	Spectral	0.9066	0.8010	0.7215	0.9019
	EM	0.8884	0.8873	0.8959	0.9015

Table 1: Comparison of AUC scores and running time for EM and spectral algorithm for various model parameters

funct mid-western airline company. For each flight, the data has a record of 186 parameters, sampled at 1 Hz, including sensor readings and pilot actions. We considered a problem of anomaly detection in aviation systems [11] and used HSMM to detect abnormal flights based on pilot actions. Specifically, we modeled the phases of the flight as hidden states and the pilot actions are the observations from these phases (see [13] for more details). We focused on a part of flight related to approach ($15 - 60$ minutes in duration) for a subset of flights landing at the same airport. We chose 9 pilot commands, among which are "selected altitude", "selected heading", etc.

A simple data filter, based on the histogram of the pilot actions, was applied to select 10020 normal flights for training. A test set contained 200 flights, with 100 of them being similar to the training set and the rest were selected from the flights rejected by the filter. Most of abnormal flights contained low occurrence events, such as fast descent, unusual usage of air brakes, etc., and few significant anomalies, e.g., the aborted descent in order to delay the flight. The length of the considered sequences varied anywhere from 500 to 4000 time stamps.

We applied EM and spectral algorithm to compute the normalized joint log-likelihood of the observed pilot actions, Figure 5 shows the results. The high-likelihood sequences were considered normal and low-likelihood ones classified as anomalous (see plots (a) and (b)). Both algorithms achieved similar detection accuracy, with the spectral algorithm having the Area Under Curve (AUC) score of 0.91 and the EM had AUC = 0.89. On the other hand, the computational time of the spectral algorithm was orders of magnitude smaller as compared to EM (see third column in Table 1). We also compared performance of both algorithm on the same flight data while varying the dimensionality of the HSMM parameters (see Table 1). We can see that although the performance of EM and spectral algorithm is similar across many models, the latter offers significant computational savings.

7 Conclusion

In this paper, we present a novel spectral algorithm to perform inference in HSMM. Our approach is based on estimating certain sample moments of size logarithmic in maximum state persistence and requires fixed number of matrix multiplications and inverses, independent of sequence length. Empirical evaluation on synthetic and real datasets illustrate the promise of the proposed spectral algorithm, especially for large datasets. Going forward, we plan to explore if similar spectral methods can be developed for inference in more general dynamic Bayesian networks.

Acknowledgements

We thank reviewers for helpful comments and suggestions. The research was supported by NSF grants IIS-1447566, IIS-1422557, CCF-1451986, CNS-1314560, IIS-0953274, IIS-1029711, and by NASA grant NNX12AQ39A. Authors acknowledge technical support from the University of Minnesota Supercomputing Institute.

References

[1] NASA Flight Dataset. Available at https://c3.nasa.gov/dashlink/projects/85/.

[2] A. Anandkumar, R. Ge, D. Hsu, and S. M. Kakade. A tensor spectral approach to learning mixed membership community models. In *Conference on Learning Theory*, 2013.

[3] A. Anandkumar, A. Javanmard, D. Hsu, and S. M. Kakade. Learning linear bayesian networks with latent variables. In *Proceedings of the International Conference on Machine Learning*, volume 28, pages 249–257, 2013.

[4] D. Barber. *Bayesian Reasoning and Machine Learning*. Cambridge University Press, 2012.

[5] L. E. Baum and T. Petrie. Statistical inference for probabilistic functions of finite state Markov chains. *The Annals of Mathematical Statistics*, 37(6):1554–1563, 1966.

[6] S. Chiappa. Explicit-duration Markov switching models. *Foundations and Trends in Machine Learning*, 7(6):803–886, 2014.

[7] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, pages 1–38, 1977.

[8] D. Hsu, S. M. Kakade, and T. Zhang. A spectral algorithm for learning hidden Markov models. *Journal of Computer and System Sciences*, 78(5):1460 – 1480, 2012.

[9] H. A. Kiers. Towards a standardized notation and terminology in multiway analysis. *Journal of chemometrics*, 14(3):105–122, 2000.

[10] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, 2009.

[11] B. Matthews, S. Das, K. Bhaduri, K. Das, R. Martin, and N. Oza. Discovering anomalous aviation safety events using scalable data mining algorithms. *Journal of Aerospace Information Systems*, 10(10):467–475, 2013.

[12] I. Melnyk and A. Banerjee. A Spectral Algorithm for Inference in Hidden Semi-Markov Models. *ArXiv e-prints, arXiv:1407.3422*, 2014.

[13] I. Melnyk, P. Yadav, M. Steinbach, J. Srivastava, V. Kumar, and A. Banerjee. Detection of precursors to aviation safety incidents due to human factors. In *Workshop on Domain Driven Data Mining (in conjunction with ICDM 2013)*, pages 407–412, 2013.

[14] E. Mossel and S. Roch. Learning nonsingular phylogenies and hidden Markov models. In *Proceedings of the Annual ACM Symposium on Theory of Computing*, pages 366–375, 2005.

[15] K. P. Murphy. Hidden semi-Markov models. Available at http://www.cs.ubc.ca/ murphyk/Papers/segment.pdf. 2002.

[16] A. Parikh, L. Song, M. Ishteva, G. Teodoru, and E. Xing. A spectral algorithm for latent junction trees. In *Proceedings of the 28th Conference Annual Conference on Uncertainty in Artificial Intelligence*, pages 675–684, 2012.

[17] A. Parikh, L. Song, and E. Xing. A spectral algorithm for latent tree graphical models. In *Proceedings of the 28th International Conference on Machine Learning*, pages 1065–1072, 2011.

[18] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[19] X. Tan and H. Xi. Hidden semi-Markov model for anomaly detection. *Applied Mathematics and Computation*, 205(2):562 – 567, 2008.

[20] T.L.M. van Kasteren, G. Englebienne, and B. J. A. Krose. Activity recognition using semi-Markov models on real world smart home datasets. *Journal of ambient intelligence and smart environments*, 2(3):311–325, 2010.

[21] Y. Xie and S.-Z. Yu. A large-scale hidden semi-Markov model for anomaly detection on user browsing behaviors. *IEEE/ACM Transactions on Networking*, 17(1):54–65, 2009.

[22] S.-Z. Yu. Hidden semi-Markov models. *Artificial Intelligence*, 174(2):215 – 243, 2010.

[23] S.-Z. Yu and H. Kobayashi. An efficient forward-backward algorithm for an explicit-duration hidden Markov model. *IEEE Signal Processing Letters*, 10(1):11–14, 2003.

[24] H. Zen, K. Tokuda, T. Masuko, T. Kobayasih, and T. Kitamura. A hidden semi-Markov model-based speech synthesis system. *Transactions on Information Systems*, 90(5):825–834, 2007.