Journal of Machine Learning Research 18 (2017) 1-39 Submitted 9/15; Revised 3/17; Published 4/17

A Spectral Algorithm for Inference in Hidden semi-Markov Models

Igor Melnyk IGOR.MELNYK@IBM.COM
IBM T. J. Watson Research Center
Yorktown Heights, NY 10598, USA

Arindam Banerjee BANERJEE@CS.UMN.EDU
Department of Computer Science and Engineering
University of Minnesota
Minneapolis, MN 55414, USA

Editor: Le Song

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. Unlike expectation maximization (EM), our approach correctly estimates the probability of given observation sequence based on a set of training sequences. Our approach is based on estimating moments from the sample, whose number of dimensions depends only logarithmically on the maximum length of the hidden state persistence. Moreover, the algorithm requires only a few matrix inversions and is therefore computationally efficient. Empirical evaluations on synthetic and real data demonstrate the advantage of the algorithm over EM in terms of speed and accuracy, especially for large data sets.

Keywords: Graphical models, hidden semi-Markov model, spectral algorithm, tensor analysis, aviation safety

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) (Chiappa, 2014; Murphy, 2002; Yu, 2010). 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. The ability to flexibly model such latent state persistence turns out to be useful in many application areas, including anomaly detection (Tan and Xi, 2008; Xie and Yu, 2009), activity recognition (van Kasteren et al., 2010), and speech synthesis (Zen et al., 2007). Such state persistence is in contrast to HMMs, which use a Markov chain over latent state transitions and hence have an implicit geometric distribution for the state duration (Rabiner, 1989).

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 HSMMs usually follow the initial idea due to Rabiner (Rabiner, 1989) based on the modifications of the Baum-Welch algorithm (Baum and Petrie, 1966), which are all variants of the expectation maximization (EM) framework, presented in (Dempster et al., 1977). Once the parameters are estimated, we can then perform inference using, e.g., the forward-backward algorithm of (Yu and Kobayashi, 2003). However, since EM, in general, has no global guarantees in estimating the parameters correctly and can suffer from slow convergence, such methods can be inefficient and/or inconsistent.

Bayesian nonparametric approaches based on hierarchical Dirichlet processes have also been proposed for HMMs (Fox et al., 2008) and HSMMs (Johnson and Willsky, 2013). Such models avoid the need to specify the number of latent states and can, in principle, learn it from data. However, in practice, inference algorithms for such models are often sensitive to initialization and may suffer from slow convergence.

In recent years, there has been an increased interest in spectral algorithms, which provide computationally efficient, local-minimum-free, provably consistent inference and/or parameter estimation algorithms for latent variable models. For example, (Anandkumar et al., 2013a, 2014b, 2013c) 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. The main idea is based on a tensor decomposition of certain low order moments, computable directly from data, in order to extract the model parameters.

In many problems, however, the end goal is not the recovery of model parameters but statistical inference, i.e., computing the probability of a given test sequence, which may be doable without estimating the canonical model parameters. In this regard, (Hsu et al., 2012) have proposed an efficient spectral algorithm for inference in HMMs. It is based on the idea of expressing the probability of the observed sequence in a representation which does not depend on the model parameters and uses easily computable second and third order sample moments to perform inference. Although their work has been used in models on sequences and trees used in Natural Language Processing (NLP) and Reinforcement Learning (RL) (Boots and Gordon, 2010; Dhillon et al., 2011; Balle et al., 2011; Cohen et al., 2014), their approach is not easily extendable to general latent variable models. The work of (Parikh et al., 2011), on the other hand, introduced a spectral algorithm to perform inference in latent tree graphical models with arbitrary topology, and later in (Parikh et al., 2012) a general spectral inference framework for latent junction trees.

In this paper, we utilize the framework of (Parikh et al., 2012) and introduce a novel spectral algorithm for inference in HSMMs. Since we address a more specific problem than (Parikh et al., 2012), 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 (Parikh et al., 2012). There are two main technical contributions in this work:

- By exploiting the *homogeneity* of HSMMs we make our proposed algorithm more efficient and accurate than the algorithm which directly follows from the recipe in (Parikh et al., 2012) for general graphs. In particular, our approach ensures that during the training phase the number of matrix multiplications and inverses is fixed and independent of the sequence length of the observations.

- Through careful analysis we show that the number of dimensions in the sample moments (represented as a multidimensional matrix or a tensor) in estimated observable representation depends only *logarithmically* on the maximum length of latent state persistence (this is in contrast to a standard implementation, which would have a linear dependence).

In experiments, comparing our method with EM on both synthetic and real data sets, two observations stand out: (i) the spectral method gets similar or better performance than EM as the number of samples increases, and (ii) 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. The idea of the observable representations was first introduced with the name 'observable operators models' by (Jaeger, 2000) in the context of constructing learning algorithm for the identification of linearly dependent processes. Our formulation cannot be directly used to infer hidden states, although methods such as in (Mossel and Roch, 2005) can be potentially utilized to recover original HSMM parameters from the learned representation. Finally, we note that the similar ideas of using homogeneity of HMMs to improve algorithm's efficiency has also been utilized in other related works, e.g., (Siddiqi et al., 2010; Hsu et al., 2012).

The rest of the paper is organized as follows: We introduce notation in Section 2. In Section 3, we present HSMM inference from a tensor product perspective and in Section 4 introduce the spectral algorithm for inference. In Section 5, we present a careful technical analysis to establish logarithmic dependence of the number of modes in the tensor on maximum latent state persistence. We present experimental results in Section 6 and conclude in Section 7.

2. Notation and Preliminaries

In this section, we cover basic facts about tensor algebra. Detailed tutorials on tensors can be found in (Kiers, 2000) or (Kolda and Bader, 2009). A tensor is defined as a multidimensional array of data, which will be denoted by boldface Euler script letters, e.g., $\underset{m_1,\ldots,m_N}{\boldsymbol{\mathcal{X}}} \in \mathbb{R}^{I_{m_1}\times\cdots\times I_{m_N}}$, which is N-mode tensor of dimensions $I_{m_1} \times \cdots \times I_{m_N}$. A specific mode is denoted by the subscript variable m_i, whose dimension is I_{m_i}.

Any tensor can be matrisized (or flattened) into a matrix. This mapping can be done in multiple ways, the only requirement is that the number of elements is preserved and the mapping is one-to-one. 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 $\boldsymbol{\mathcal{X}}$ is denoted by a corresponding capital boldface letter, e.g., $\underset{p_1,\ldots,p_K q_1,\ldots,q_L}{\mathbf{X}} \in \mathbb{R}^{I_{p_1}\cdots I_{p_K}\times I_{q_1}\cdots I_{q_L}}$.

Tensor Multiplication. Multiplication of two tensors is performed along specific modes. For this, we flatten each tensor to a matrix, perform the usual matrix multiplication and transform the result back to a tensor. The multiplication is denoted by a symbol $\times$ with an optional subscript representing the modes along which the operation is performed, e.g.,:

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

where $\underset{q_1,\ldots,q_L,r_1,\ldots,r_M}{\boldsymbol{\mathcal{Y}}} \in \mathbb{R}^{I_{q_1}\times\cdots\times I_{q_L}\times I_{r_1}\times\cdots\times I_{r_M}}$ and the resulting tensor on the left hand side is of the form $\underset{p_1,\ldots,p_K,r_1,\ldots,r_M}{\boldsymbol{\mathcal{Z}}} \in \mathbb{R}^{I_{p_1}\times\cdots\times I_{p_K}\times I_{r_1}\times\cdots\times I_{r_M}}$. Observe that in the above, we can flatten the tensors $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{Y}}$ in multiple different ways as long as the matrix multiplication remains valid. For example, we could assign the multiplication modes in both tensors to columns, in this case the matrix product becomes $\mathbf{Z} = \mathbf{X}\mathbf{Y}^T$. Alternatively, the tensor $\boldsymbol{\mathcal{Y}}$ could be matrisized with the multiplication modes corresponding to rows, resulting in the product $\mathbf{Z} = \mathbf{X}\mathbf{Y}$.

$\underset{p,q,r}{\mathcal{X}} \in \mathbb{R}^{I_p \times I_q \times I_r}$	N-mode tensor of dimensions $I_p \times I_q \times I_r$
$\underset{p,q,r}{\mathbf{X}} \in \mathbb{R}^{I_p I_q \times I_r}$	Matricization of tensor $\underset{p,q,r}{\mathcal{X}}$ with $I_p I_q$ rows and I_r columns
$\underset{p,q,r}{\mathcal{X}} \times_r \underset{r,s,t}{\mathcal{Y}}$	Multiplication of tensor $\underset{p,q,r}{\mathcal{X}}$ and tensor $\underset{r,s,t}{\mathcal{Y}}$ along mode r
$\underset{p,q,r}{\mathcal{X}} \times_{q,r} \underset{p,q,r}{\mathcal{X}}^{-1} = \underset{p,p}{\mathfrak{I}} = \underset{p}{\mathfrak{I}}$	Inversion of tensor $\underset{p,q,r}{\mathcal{X}}$ with respect to modes q and r
$\mathbb{X}_t$	Representation of a clique in a Junction tree
$o_t \in \{1, \ldots, n_o\}$	Observation variable in HSMM
$x_t \in \{1, \ldots, n_x\}$	Latent state variable in HSMM
$d_t \in \{1, \ldots, n_d\}$	Latent duration variable in HSMM
$\mathbf{O}_{R_t} := \{o_{t+1}, o_{t+2}, \ldots\}$	Set of observations to the right of time step t
$\mathbf{O}_{L_t} := \{\ldots, o_{t-2}, o_{t-1}\}$	Set of observations to the left of time step t

Table 1: Summary of some of the key notations used throughout the paper.

An important fact about tensor multiplication is that in a series of tensor multiplications the order is irrelevant (i.e., it is an associative operation) as long as the multiplication is performed along the matching modes, e.g,

$$\underset{sp}{\mathcal{X}} \times_s \left(\underset{tr}{\mathcal{Y}} \times_r \underset{rs}{\mathcal{Z}} \right) = \left(\underset{sp}{\mathcal{X}} \times_s \underset{rs}{\mathcal{Z}} \right) \times_r \underset{tr}{\mathcal{Y}}.$$

If we let the matrisized tensors to be $\mathbf{X} \in \mathbb{R}^{I_p \times I_s}$, $\mathbf{Y} \in \mathbb{R}^{I_t \times I_r}$ and $\mathbf{Z} \in \mathbb{R}^{I_r \times I_s}$, then the above can be verified to be true since

$$\mathbf{X} \left(\mathbf{Y}\mathbf{Z} \right)^T = \left(\mathbf{X}\mathbf{Z}^T \right) \mathbf{Y}^T.$$

To reduce clutter, in many places we will drop the multiplication subscripts. The implied modes of multiplication can then be inferred from the subscripts of the tensors. Specifically, when two tensors are multiplied, we first check their modes and then multiply along the modes which are common to both of them. For example, in the product $\underset{pqr}{\mathcal{X}} \times \underset{qsr}{\mathcal{Y}}$, the implied multiplication is performed along the common modes, i.e., q and r.

Tensor Inversion. We also discuss the operation of tensor inversion. Tensor inverse $\mathcal{X}^{-1}$ is always defined with respect to a certain subset of modes and can be written as follows:

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

where the inversion is performed along the modes $q_1, \ldots, q_L$, and $\underset{p_1,\ldots,p_K,p_1,\ldots,p_K}{\mathfrak{I}}$ denotes an identity tensor, whose elements are everywhere zero, except $\mathfrak{I}(i_1, \ldots, i_K, i_1, \ldots, i_K) = 1$. To perform inversion, we first convert tensor to a matrix, i.e., matrisized tensor. 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. Obviously, for the full rank square matrices both choices would produce the same result. For example, in the above equation the matrisized tensor might be of the form $\underset{p_1,\ldots,p_K q_1,\ldots,q_L}{\mathbf{X}} \in \mathbb{R}^{I_{p_1} \cdots I_{p_K} \times I_{q_1} \cdots I_{q_L}}$, therefore, we would compute the right matrix inverse so that the modes $q_1, \ldots, q_L$ are eliminated. If the matrisized $\mathbf{X}$ has full row rank, then the inverse can be computed, otherwise we could only compute its pseudo-inverse. Tensorizing the matrix $\mathbf{X}^{-1}$ gives us the desired tensor inverse.

Mode Duplication. Observe that in the above, the tensor $\underset{p_1,\ldots,p_K,p_1,\ldots,p_K}{\mathcal{I}}$ has duplicate modes. In general, if a tensor has duplicate modes, the corresponding sub-tensor can be interpreted as a hyper-diagonal. For example, if for a tensor $\underset{pq}{\mathcal{X}}$ we construct a tensor $\underset{pppq}{\overline{\mathcal{X}}}$, which has its mode p duplicated three times, then for a fixed index i, the sub-tensor $\overline{\mathcal{X}}(:,:,:,i)$ is a hypercube with elements $\mathcal{X}(:,i)$ on the diagonal.

Mode duplication enables us to multiply several tensors along the same mode. For example, if we need to multiply tensors $\underset{sp}{\mathcal{X}}$, $\underset{pr}{\mathcal{Y}}$ and $\underset{tp}{\mathcal{Z}}$ along the mode p, then a simple product of the form

$$\underset{sp}{\mathcal{X}} \times_p \underset{pr}{\mathcal{Y}} \times_p \underset{tp}{\mathcal{Z}}$$

cannot be done since any product of two tensors along the mode p would eliminate it, preventing any further multiplications. In general, if there are N multiplications along the specific mode, then there are must be cumulatively $2N$ number of times such a mode is encountered in the participating tensors. In our example, we might duplicate the mode p in, say, tensor $\mathcal{Z}$ to have

$$\underset{sp}{\mathcal{X}} \times_p \left(\underset{pr}{\mathcal{Y}} \times_p \underset{tpp}{\mathcal{Z}} \right) = \underset{sp}{\mathcal{X}} \times_p \underset{prt}{\mathcal{W}} = \underset{srt}{\mathcal{V}},$$

so that there are two multiplications over mode p and cumulatively there are four times such a mode is encountered in the participating tensors. To reduce clutter, we sometimes do not explicitly show the duplicated variables in the subscripts; the implied mode repetition will be evident from the context or explicitly stated in cases when there is a confusion. For example, the identity tensor will often be written as $\underset{p_1,\ldots,p_K}{\mathcal{I}}$.

Tensor rank Finally, we discuss the meaning of a tensor rank. A tensor can have multiple ranks and each of them is defined based on the rank of a particular matricization. For example, consider a tensor $\underset{pqs}{\mathcal{X}}$. If we flatten it to a matrix $\mathbf{X} \in \mathbb{R}^{I_p \times I_q I_s}$ then it can have a rank r_1. On the other hand, a matricization of the form $\mathbf{X} \in \mathbb{R}^{I_p I_q \times I_s}$ can have a rank r_2, and so on. In our derivations, the particular rank we are referring to will be evident from the context.
In Table 2 we summarized some of the key notations used throughout the paper.

3. Problem Formulation

In this paper, we consider the problem of inference in HSMM[1] (see Figure 1). Unlike the popular HMM, which has a geometric probability for state persistence, i.e., the probability of persisting in the same state over t time steps decreases as π^t, where π is the probability of persistence for one time step, HSMM explicitly models state persistence. 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

[1] To reduce clutter, in the main paper we only consider the model for a general time stamp t and ignore the initial ($t = 0$) and final ($t = T$) steps of the model, whose representation differs slightly from what is shown in Figure 1. The details for these parts are presented separately in Appendix B.

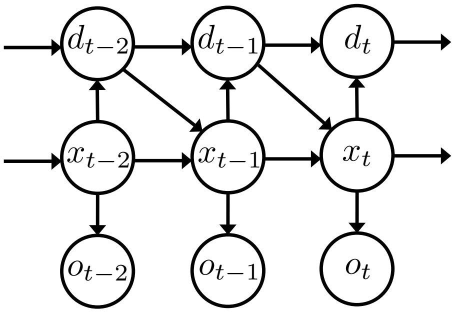

Figure 1: Hidden Semi-Markov Model (HSMM) depicted as a dynamic Bayesian network. Here $o_t \in \{1, \dots, n_o\}$ denotes an observation at time step t, $x_t \in \{1, \dots, n_x\}$ is a latent state and $d_t \in \{1, \dots, n_d\}$ is the length of state persistence at time step t.

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

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

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

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 $d_t|x_t$. For our analysis, we assume $p(d_t|x_t, d_{t-1} = 1)$ to be a discrete 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 discrete sequences $\{\mathbf{S}^1, \dots, \mathbf{S}^N\}$ drawn independently from the HSMM model, where each sequence is defined as $\mathbf{S}^i = \{o_1^i, \dots, o_{T_i}^i\}, i = 1, \dots, N$, our goal is to compute the probability $p(\mathbf{S}^{test})$ of any given test sequence $\mathbf{S}^{test} = (o_1^{test}, \dots, o_T^{test})$. A traditional approach would be to estimate the CPTs using the EM algorithm, and use the estimates to compute $p(\mathbf{S}^{test})$. However, the EM algorithm is not guaranteed to estimate the parameters optimally, and hence the computation of $p(\mathbf{S}^{test})$ may be incorrect. The focus of our work is to develop a provably correct spectral algorithm for computing the probability $p(\mathbf{S}^{test})$.

3.1 HSMM in Tensor Notations

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 $X \in \mathbb{R}^{n_x \times n_x}$, 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:

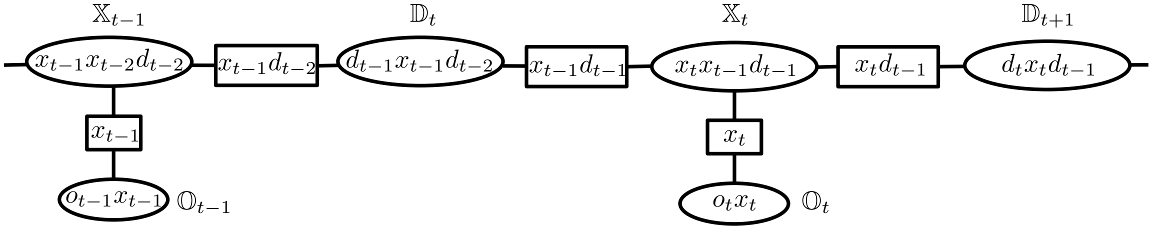

Figure 2: Junction Tree for Hidden Semi-Markov Model. The ovals represent cliques, which are denoted by capital blackboard bold variables; the rectangles denote separators. Symbols within the shapes represent the variables on which the corresponding potentials depend.

Assumptions

$A1$. X is full rank and has non-zero probability of visiting any state from any other state.

$A2$. D has a non-zero probability of any duration in any state.

$A3$. O is full column rank and, as a consequence, $n_x \leq n_o$.

We provide some comments on the above assumptions. We note that the assumption $A1$ can be relaxed to allow zero entries (while still ensuring full rank structure) and thus prevent certain states to be directly reachable from other states; however, this would require more involved analysis based on the mixing time of the corresponding Markov chain (Levin et al., 2009), and is not pursued in this work. Also, observe that the assumption of $n_x \leq n_o$ is needed in order to ensure that hidden states are identifiable, although recent work is showing that such an assumption can be relaxed in some cases (Bailly et al., 2009; Anandkumar et al., 2013b). Intuitively, it means that the number of different observations coming from each state is large enough, so that one hidden state can be differentiated from the other.

To express the joint probability $p(o_1, \ldots, o_T)$ for any possible observation sequence in tensor form, we utilize the junction tree algorithm (Barber, 2012). The resulting tree is shown in Figure 2 and it corresponds to the graphical model of HSMM in Figure 1. Recall, that the junction tree is a tree-structured representation of an arbitrary graph enabling efficient inference. It can be constructed by forming a maximal spanning tree from the cliques of the graph. The cliques then represent vertices in the junction tree and the edges connecting the vertices are labeled with variables common to two cliques it connects. The set of variables on the edges are referred to as separators. For example, in Figure 2 the cliques $\mathbb{X}_t$ and $\mathbb{D}_t$ have two variables in common, x_{t-1} and d_{t-1}, and which define the sepatator between $\mathbb{X}_t$ and $\mathbb{D}_t$.

We proceed by representing the clique CPTs of the junction tree as tensors. For example, the clique $\mathbb{X}_t$, containing the CPT of $p(x_t|x_{t-1}, d_{t-1})$ is represented as 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. We also keep the conditioning symbol | for clarity. Similarly, we represent the clique $\mathbb{D}_t$ with its CPT $p(d_t|x_t, d_{t-1})$ as tensor $\underset{d_t|x_td_{t-1}}{\mathcal{D}}$, and $\mathbb{O}_t$ containing $p(o_t|x_t)$ as tensor $\underset{o_t|x_t}{\mathcal{O}}$.

If we denote the joint probability of the observed sequence $p(o_1, \ldots, o_T)$ as $\underset{o_1,\ldots,o_T}{\mathcal{P}}$ then the message passing for the junction tree algorithm in Figure 2 can be represented as tensor multiplications:

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

where, for simplicity, we denoted by $\prod_t$ the tensor product over multiple time steps.

Note that in (3) the neighboring tensors are multiplied along the modes which are the separator variables between two corresponding neighboring cliques in Figure 2. Therefore, as we discussed in Section 2, if a certain mode of a tensor is to participate multiple times in products with other tensor, the mode must be duplicated for the expression to remain correct. It can easily be seen from the junction tree that the number of times the mode is duplicated depends on the number of times such a variable appears in separators adjacent to the clique. For example, the tensor $\underset{x_t x_t|x_{t-1}d_{t-1}d_{t-1}}{\mathcal{X}}$ has a mode x_{t-1} appearing once in the separator connecting $\mathbb{X}_t$ and $\mathbb{D}_t$ in Figure 2, while x_t appears a total of two times, once in the separator connecting $\mathbb{X}_t$ and $\mathbb{O}_t$, and once in the separator connecting $\mathbb{X}_t$ and $\mathbb{D}_{t+1}$. Finally, d_{t-1} appears in the separator between $\mathbb{D}_t$ and $\mathbb{X}_t$, and between $\mathbb{D}_{t+1}$ and $\mathbb{X}_t$. Applying the same reasoning to tensors $\mathcal{D}$ and $\mathcal{O}$ results in the expression (3).

3.2 Summary of Technical Results

In this work, we represent expression (3), which is defined in terms of unknown model parameters, in a different *observable form*, where all the factors can be estimated directly from data using certain sample moments without knowledge of model parameters. Such an observable form is derived in Sections 4.1 and 4.2. Based on the observable form, in Section 4.3 we propose a simple spectral algorithm which requires estimating $\mathcal{X}$, $\mathcal{D}$ and $\mathcal{O}$ for all the time stamps t. This estimation process is expensive as it involves costly tensor operations to be performed at each time index t. Moreover, the accurate estimation of these tensors requires large number of training sequences which might not be available, leading to inaccurate and unstable computations. However, exploiting the homogeneity property of HSMMs, i.e., the probability distributions represented by the above tensors are the same across all time t, we derive a computationally efficient and accurate spectral algorithm in Section 4.4 which requires the estimation of only three tensors for all the time stamps t. Although the computational complexity of the inference, i.e., the evaluation of expression (3), is not affected by the introduced modifications, the overall algorithm becomes faster and more accurate.

In Section 5 we return to the results of Sections 4.1 and establish the conditions under which the derived observable form can be computed from data. In particular, our analysis shows that the number of dimensions of the required sample moments (in the form of tensors, estimated from data and representing the co-occurrence frequency of certain observable variables), has logarithmic dependence on the longest state persistence n_d. Such conclusion is in contrast to the analysis, which would follow from the work of (Parikh et al., 2012), in which case the required number of dimensions in the estimated sample moments would have had linear dependence on n_d. The exponential reduction in the size of the estimated tensors represents significant improvement in algorithm's efficiency and accuracy since the multidimensional matrices are of smaller size and consequently more data is available to estimate each of its entry.

4. Spectral Algorithm for Inference in HSMM

In this Section we present the details of the spectral inference approach. In particular, in Sections 4.1 and 4.2 we derive observable tensor representation and show how to estimate each of its factors directly from data. Practical algorithms implementing these ideas are then derived in Sections 4.3 and 4.4.

4.1 Observable Tensor Representation

Observe that the computation of the joint probability in (3) requires knowledge of the unknown model parameters. Our goal is to change the tensor representation such that $\underset{o_1,\ldots,o_T}{\mathfrak{P}}$ can be written in terms of the quantities directly computable from data. To that end, we follow (Parikh et al., 2012) and between every two factors in (3) introduce an identity tensor with the modes corresponding to the modes along which the multiplication is performed. For example, consider a part of (3) after introducing identity tensors:

$$\times \underset{x_{t-1}d_{t-2}}{\mathfrak{I}} \times_{x_{t-1}d_{t-2}} \underset{d_{t-1}|x_{t-1}x_{t-1}d_{t-2}}{\mathfrak{D}} \times_{x_{t-1}d_{t-1}} \underset{x_{t-1}d_{t-1}}{\mathfrak{I}} \times_{x_{t-1}d_{t-1}} \left(\underset{x_t x_t|x_{t-1}d_{t-1}d_{t-1}}{\mathfrak{X}} \times_{x_t} \underset{x_t}{\mathfrak{I}} \times_{x_t} \underset{o_t x_t}{\mathfrak{O}} \right) \times_{x_t d_{t-1}} \underset{x_t d_{t-1}}{\mathfrak{I}} \times, \quad (4)$$

where all the identity tensors have duplicated modes which are not shown.

Now rewrite each of the identity tensors in (4) as a multiplication of some factor times its inverse. For example,

$$\underset{x_t}{\mathfrak{I}} = \underset{\omega_{x_t}x_t}{\mathfrak{F}} \times_{\omega_{x_t}} \underset{\omega_{x_t}x_t}{\mathfrak{F}^{-1}},$$

for some invertible factor $\underset{\omega_{x_t}x_t}{\mathfrak{F}}$, whose modes are x_t and ω_{x_t}. Note that the choice of mode x_t is fixed and is determined by the modes of the identity tensor $\underset{x_t}{\mathfrak{I}}$, while the mode ω_{x_t} is not fixed and we have freedom in selecting it as convenient. Moreover, observe that since the tensor inversion is done along the mode ω_{x_t} and the matrix $\mathbf{F}$ has its rows associated with mode ω_{x_t}, we need to ensure such a matrix has full column rank for the inverse to exist and for the product $\mathbf{F}^{-1}\mathbf{F}$ to be the identity matrix (see Section 2 for more details on tensor inversion). Based on the above discussion, we choose tensor $\mathfrak{F}$ such that (i) ω_{x_t} are the observed variables, (ii) $\underset{\omega_{x_t}x_t}{\mathfrak{F}}$ is invertible, i.e., matrix $\mathbf{F}$, whose columns correspond to x_t, has full column rank, and (iii) we interpret the factor $\underset{\omega_{x_t}x_t}{\mathfrak{F}}$ as corresponding to a conditional probability distribution, i.e., $p(\omega_{x_t}|x_t)$ and therefore write $\underset{\omega_{x_t}|x_t}{\mathfrak{F}}$.

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 three modified tensors:

$$\begin{aligned}
\underset{\omega_{x_{t-1}d_{t-2}}\omega_{x_{t-1}d_{t-1}}}{\tilde{\mathfrak{D}}} &= \underset{\omega_{x_{t-1}d_{t-2}}|x_{t-1}d_{t-2}}{\mathfrak{F}^{-1}} \times_{x_{t-1}d_{t-2}} \underset{d_{t-1}|x_{t-1}x_{t-1}d_{t-2}}{\mathfrak{D}} \times_{x_{t-1}d_{t-1}} \underset{\omega_{x_{t-1}d_{t-1}}|x_{t-1}d_{t-1}}{\mathfrak{F}} \\
\underset{\omega_{x_{t-1}d_{t-1}}\omega_{x_t}\omega_{x_t d_{t-1}}}{\tilde{\mathfrak{X}}} &= \underset{\omega_{x_{t-1}d_{t-1}}|x_{t-1}d_{t-1}}{\mathfrak{F}^{-1}} \times_{x_{t-1}d_{t-1}} \left(\underset{x_t x_t|x_{t-1}d_{t-1}d_{t-1}}{\mathfrak{X}} \times_{x_t} \underset{\omega_{x_t}|x_t}{\mathfrak{F}} \right) \times_{x_t d_{t-1}} \underset{\omega_{x_t d_{t-1}}|x_t d_{t-1}}{\mathfrak{F}} \\
\underset{\omega_{x_t}o_t}{\tilde{\mathfrak{O}}} &= \underset{\omega_{x_t}|x_t}{\mathfrak{F}^{-1}} \times_{x_t} \underset{o_t|x_t}{\mathfrak{O}} .
\end{aligned}$$

Note that although each of the above tensors depends only on certain observed variables ω, for a concrete algorithm one has to decide what these ω are, and also how to estimate the associated tensors from data. The right hand side in the above expressions depend on the unknown model parameters, whereas the tensors on the left do not correspond to valid probability distributions (due to the presence of inverses $\mathfrak{F}^{-1}$), and so cannot be estimated from data using sample moments. For example, $\underset{\omega_{x_{t-1}d_{t-2}}\omega_{x_{t-1}d_{t-1}}}{\tilde{\mathfrak{D}}}$ 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 $\mathfrak{F}$. From Figure 2 we can see that there are three types of separators which depend on $x_{t-1}d_{t-1}$, $x_t d_{t-1}$ and x_t, consequently, there are three types of identity tensors which we introduced in (4), i.e., $\underset{x_{t-1}d_{t-1}}{\mathfrak{I}}$, $\underset{x_t d_{t-1}}{\mathfrak{I}}$ and $\underset{x_t}{\mathfrak{I}}$. 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 are multiple choices for these sets, one of them is $\omega_{x_{t-1}d_{t-1}} = \omega_{x_t d_{t-1}} = \{o_{t+1}, o_{t+2}, \dots\}$ for all t (see Figure 3 for an illustration). Ideally, we want these sets to be of minimal size, since they need to be estimated from observations. The detailed description of how many and which of these observations to select to get a minimal set is deferred until Section 5, where we also show that we can set $\omega_{x_t} = o_t$.

In what follows, we define $\mathbf{O}_{R_t} := \{o_{t+1}, o_{t+2}, \dots, o_{t+\tau}\}$ (see Figure 3) to emphasize that this is a fixed set of observations whose length τ is yet to be determined, starting after time stamp t and going to the right (or forward in time) in the graphical model in Figure 1. With these definitions, setting $\omega_{x_{t-1}d_{t-1}} = \mathbf{O}_{R_t}$, $\omega_{x_t d_{t-1}} = \mathbf{O}_{R_t}$, $\omega_{x_{t-1}d_{t-2}} = \mathbf{O}_{R_{t-1}}$ and $\omega_{x_t} = o_t$, we can now rewrite (3) in the form:

$$\underset{o_1,\dots,o_T}{\mathcal{P}} = \prod_t \underset{\mathbf{O}_{R_{t-1}}\mathbf{O}_{R_t}}{\tilde{\mathfrak{D}}} \times_{\mathbf{O}_{R_t}} \left(\underset{\mathbf{O}_{R_t} o_t \mathbf{O}_{R_t}}{\tilde{\mathfrak{X}}} \times_{o_t} \underset{o_t o_t}{\tilde{\mathfrak{O}}} \right). \tag{5}$$

Comparing (3) and (5) 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 for the model parameters.

4.2 Estimation of Observable Tensors

In this Section we express each of the tensors in (5) in forms which can be directly estimated from the observed sequences.

4.2.1 COMPUTATION OF TENSOR $\underset{\mathbf{O}_{R_{t-1}}\mathbf{O}_{R_t}}{\tilde{\mathfrak{D}}}$

Consider the tensor from Section 4.1

$$\underset{\mathbf{O}_{R_{t-1}}\mathbf{O}_{R_t}}{\tilde{\mathfrak{D}}} = \underset{\mathbf{O}_{R_{t-1}}|x_{t-1}d_{t-2}}{\mathfrak{F}^{-1}} \times_{x_{t-1}d_{t-2}} \underset{d_{t-1}|x_{t-1}x_{t-1}d_{t-2}}{\mathfrak{D}} \times_{x_{t-1}d_{t-1}} \underset{\mathbf{O}_{R_t}|x_{t-1}d_{t-1}}{\mathfrak{F}}, \tag{6}$$

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 3):

$$p(\mathbf{O}_{L_{t-1}}, \mathbf{O}_{R_{t-1}}) = \sum_{x_{t-1}d_{t-2}} p(\mathbf{O}_{L_{t-1}}|x_{t-1}d_{t-2}) p(\mathbf{O}_{R_{t-1}}|x_{t-1}d_{t-2}) p(x_{t-1}d_{t-2}). \tag{7}$$

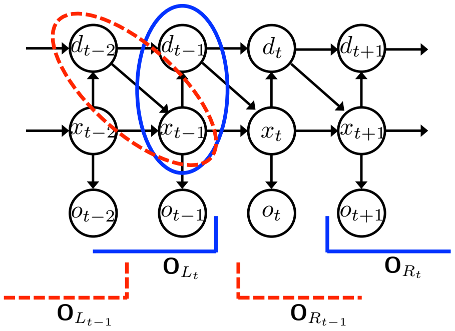

Figure 3: Conditional independence in HSMM. The figure depicts two sets of relationships: $\mathbf{O}_{L_t}$ and $\mathbf{O}_{R_t}$ are independent conditioned on $x_{t-1}d_{t-1}$, similarly, $\mathbf{O}_{L_{t-1}}$ and $\mathbf{O}_{R_{t-1}}$ are conditionally independent given $x_{t-1}d_{t-2}$. We defined $\mathbf{O}_{L_t} = \{\ldots, o_{t-2}, o_{t-1}\}$ and $\mathbf{O}_{R_t} = \{o_{t+1}, o_{t+2}, \ldots\}$.

The above conditional independence relationship can be written in tensor form:

$$\underset{\mathbf{O}_{L_{t-1}}\mathbf{O}_{R_{t-1}}}{\mathcal{M}} = \underset{\mathbf{O}_{L_{t-1}}|x_{t-1}d_{t-2}}{\mathcal{F}} \times_{x_{t-1}d_{t-2}} \underset{\mathbf{O}_{R_{t-1}}|x_{t-1}d_{t-2}}{\mathcal{F}} \times_{x_{t-1}d_{t-2}} \underset{x_{t-1}d_{t-2}}{\mathcal{K}}, \tag{8}$$

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 (see the discussion on mode duplication in Section 2). 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 3).

Next, we express the inverse of the tensor $\underset{\mathbf{O}_{R_{t-1}}|x_{t-1}d_{t-2}}{\mathcal{F}}$ from (8) and substitute back to (6). For this, we observe that in (6) the tensor $\mathcal{F}^{-1}$ is inverted with respect to mode $\mathbf{O}_{R_{t-1}}$, therefore, we do the following:

$$\begin{aligned}\underset{\mathbf{O}_{L_{t-1}}\mathbf{O}_{R_{t-1}}}{\mathcal{M}} \times_{\mathbf{O}_{R_{t-1}}} \underset{\mathbf{O}_{R_{t-1}}|x_{t-1}d_{t-2}}{\mathcal{F}^{-1}} &= \underset{\mathbf{O}_{L_{t-1}}|x_{t-1}d_{t-2}}{\mathcal{F}} \times_{x_{t-1}d_{t-2}} \underset{x_{t-1}d_{t-2}}{\mathcal{I}} \times_{x_{t-1}d_{t-2}} \underset{x_{t-1}d_{t-2}}{\mathcal{K}} \\ \underset{\mathbf{O}_{R_{t-1}}|x_{t-1}d_{t-2}}{\mathcal{F}^{-1}} &= \underset{\mathbf{O}_{L_{t-1}}\mathbf{O}_{R_{t-1}}}{\mathcal{M}^{-1}} \times_{\mathbf{O}_{L_{t-1}}} \underset{\mathbf{O}_{L_{t-1}}|x_{t-1}d_{t-2}}{\mathcal{F}} \times_{x_{t-1}d_{t-2}} \underset{x_{t-1}d_{t-2}}{\mathcal{K}},\end{aligned} \tag{9}$$

where $\underset{\mathbf{O}_{L_{t-1}}\mathbf{O}_{R_{t-1}}}{\mathcal{M}^{-1}}$ is inverted with respect to mode $\mathbf{O}_{L_{t-1}}$. Next, substituting (9) back to (6), we get

$$\begin{aligned}\underset{\mathbf{O}_{R_{t-1}}\mathbf{O}_{R_t}}{\tilde{\mathcal{D}}} &= \underset{\mathbf{O}_{L_{t-1}}\mathbf{O}_{R_{t-1}}}{\mathcal{M}^{-1}} \times_{\mathbf{O}_{L_{t-1}}} \overbracket{\underset{\mathbf{O}_{L_{t-1}}|x_{t-1}d_{t-2}}{\mathcal{F}} \times_{x_{t-1}d_{t-2}} \underset{x_{t-1}d_{t-2}}{\mathcal{K}} \times_{x_{t-1}d_{t-2}} \underset{d_{t-1}|x_{t-1}x_{t-1}d_{t-2}}{\mathcal{D}} \times_{x_{t-1}d_{t-1}} \underset{\mathbf{O}_{R_t}|x_{t-1}d_{t-1}}{\mathcal{F}}} \\ &= \underset{\mathbf{O}_{L_{t-1}}\mathbf{O}_{R_{t-1}}}{\mathcal{M}^{-1}} \times_{\mathbf{O}_{L_{t-1}}} \underset{\mathbf{O}_{L_{t-1}}\mathbf{O}_{R_t}}{\mathcal{M}},\end{aligned} \tag{10}$$

where we have 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 joint probability distributions over a subset of observations $p(\mathbf{O}_{L_{t-1}}, \mathbf{O}_{R_{t-1}})$ and $p(\mathbf{O}_{L_{t-1}}, \mathbf{O}_{R_t})$, respectively, and though they are

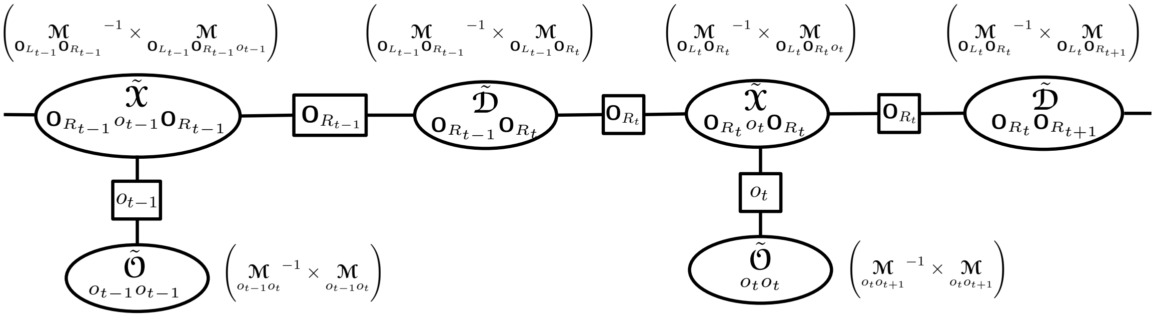
$\left(\underset{\mathbf{o}_{L_{t-1}}\mathbf{o}_{R_{t-1}}}{\mathcal{M}}^{-1} \times \underset{\mathbf{o}_{L_{t-1}}\mathbf{o}_{R_{t-1}}o_{t-1}}{\mathcal{M}}\right)$
$\left(\underset{\mathbf{o}_{L_{t-1}}\mathbf{o}_{R_{t-1}}}{\mathcal{M}}^{-1} \times \underset{\mathbf{o}_{L_{t-1}}\mathbf{o}_{R_t}}{\mathcal{M}}\right)$
$\left(\underset{\mathbf{o}_{L_t}\mathbf{o}_{R_t}}{\mathcal{M}}^{-1} \times \underset{\mathbf{o}_{L_t}\mathbf{o}_{R_t}o_t}{\mathcal{M}}\right)$
$\left(\underset{\mathbf{o}_{L_t}\mathbf{o}_{R_t}}{\mathcal{M}}^{-1} \times \underset{\mathbf{o}_{L_t}\mathbf{o}_{R_{t+1}}}{\mathcal{M}}\right)$
$\tilde{\mathcal{X}}$ $\mathbf{o}_{R_{t-1}}o_{t-1}\mathbf{o}_{R_{t-1}}$
$\mathbf{o}_{R_{t-1}}$
$\tilde{\mathcal{D}}$ $\mathbf{o}_{R_{t-1}}\mathbf{o}_{R_t}$
$\mathbf{o}_{R_t}$
$\tilde{\mathcal{X}}$ $\mathbf{o}_{R_t}o_t\mathbf{o}_{R_t}$
$\mathbf{o}_{R_t}$
$\tilde{\mathcal{D}}$ $\mathbf{o}_{R_t}\mathbf{o}_{R_{t+1}}$
o_{t-1}
o_t
$\tilde{\mathcal{O}}$ $o_{t-1}o_{t-1}$
$\left(\underset{o_{t-1}o_t}{\mathcal{M}}^{-1} \times \underset{o_{t-1}o_t}{\mathcal{M}}\right)$
$\tilde{\mathcal{O}}$ $o_t o_t$
$\left(\underset{o_t o_{t+1}}{\mathcal{M}}^{-1} \times \underset{o_t o_{t+1}}{\mathcal{M}}\right)$

Figure 4: Graphical representation of the HSMM spectral algorithm for inference in Algorithm 1. As compared to junction tree in Figure 2, the cliques and separators are now defined in terms of the tensors, which are defined with respect to the observed data. The expressions in the parenthesis show the observable representation of the corresponding tensors.

defined with respect to unknown model parameters (as, for example, in (7)), 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 observations $\{\ldots, o_{t-3}, o_{t-2}, o_{t+1}, o_{t+2}, \ldots\}$. Ideally, we want a small number of observations since we need to estimate their co-occurrence frequency from the training data. A precise characterization of how many and which of these observations suffices for the analysis will be done in Section 5.

4.2.2 COMPUTATION OF TENSOR $\underset{\mathbf{o}_{R_t}o_t\mathbf{o}_{R_t}}{\tilde{\mathcal{X}}}$

The form of this tensor was established at the beginning of Section 4.2 to be:

$$\underset{\mathbf{o}_{R_t}o_t\mathbf{o}_{R_t}}{\tilde{\mathcal{X}}} = \underset{\mathbf{o}_{R_t}|x_{t-1}d_{t-1}}{\mathcal{F}^{-1}} \times_{x_{t-1}d_{t-1}} \left(\underset{x_t x_t|x_{t-1}d_{t-1}d_{t-1}}{\mathcal{X}} \times_{x_t} \underset{o_t|x_t}{\mathcal{F}} \right) \times_{x_t d_{t-1}} \underset{\mathbf{o}_{R_t}|x_t d_{t-1}}{\mathcal{F}}. \tag{11}$$

Consider the following conditional independence relationship (see Figure 3):

$$\underset{\mathbf{o}_{L_t}\mathbf{o}_{R_t}}{\mathcal{M}} = \underset{\mathbf{o}_{L_t}|x_{t-1}d_{t-1}}{\mathcal{F}} \times_{x_{t-1}d_{t-1}} \underset{\mathbf{o}_{R_t}|x_{t-1}d_{t-1}}{\mathcal{F}} \times_{x_{t-1}d_{t-1}} \underset{x_{t-1}d_{t-1}}{\mathcal{K}}, \tag{12}$$

where $\underset{x_{t-1}d_{t-1}}{\mathcal{K}} = \underset{x_{t-1}d_{t-1}x_{t-1}d_{t-1}}{\mathcal{K}}$ and we omitted the duplicated modes.

We express the inverse of tensor $\underset{\mathbf{o}_{R_t}|x_{t-1}d_{t-1}}{\mathcal{F}}$ from the above equation

$$\underset{\mathbf{o}_{R_t}|x_{t-1}d_{t-1}}{\mathcal{F}^{-1}} = \underset{\mathbf{o}_{L_t}\mathbf{o}_{R_t}}{\mathcal{M}^{-1}} \times_{\mathbf{o}_{L_t}} \underset{\mathbf{o}_{L_t}|x_{t-1}d_{t-1}}{\mathcal{F}} \times_{x_{t-1}d_{t-1}} \underset{x_{t-1}d_{t-1}}{\mathcal{K}},$$

where tensor $\underset{\mathbf{o}_{R_t}|x_{t-1}d_{t-1}}{\mathcal{F}}$ is inverted with respect to mode $\mathbf{o}_{R_t}$, while $\underset{\mathbf{o}_{L_t}\mathbf{o}_{R_t}}{\mathcal{M}}$ is inverted with respect to mode $\mathbf{o}_{L_t}$. Substituting back to (11), we get

$$\underset{\mathbf{o}_{R_t}o_t\mathbf{o}_{R_t}}{\tilde{\mathcal{X}}} = \underset{\mathbf{o}_{L_t}\mathbf{o}_{R_t}}{\mathcal{M}^{-1}} \times_{\mathbf{o}_{L_t}} \underset{\mathbf{o}_{L_t}|x_{t-1}d_{t-1}}{\mathcal{F}} \times_{x_{t-1}d_{t-1}} \underset{x_{t-1}d_{t-1}}{\mathcal{K}} \times_{x_{t-1}d_{t-1}} \left(\underset{x_t x_t|x_{t-1}d_{t-1}d_{t-1}}{\mathcal{X}} \times_{x_t} \underset{o_t|x_t}{\mathcal{F}} \right) \times_{x_t d_{t-1}} \underset{\mathbf{o}_{R_t}|x_t d_{t-1}}{\mathcal{F}}.$$

Considering the last five factors and multiplying them together, we obtain

$$\underset{\mathbf{o}_{L_t}\mathbf{o}_{R_t}o_t}{\mathcal{M}} = \underset{\mathbf{o}_{L_t}|x_{t-1}d_{t-1}}{\mathcal{F}} \times_{x_{t-1}d_{t-1}} \underset{x_{t-1}d_{t-1}}{\mathcal{K}} \times_{x_{t-1}d_{t-1}} \left(\underset{x_t x_t|x_{t-1}d_{t-1}d_{t-1}}{\mathcal{X}} \times_{x_t} \underset{o_t|x_t}{\mathcal{F}} \right) \times_{x_t d_{t-1}} \underset{\mathbf{o}_{R_t}|x_t d_{t-1}}{\mathcal{F}} .$$

Finally, (11) can now be written as

$$\underset{\mathbf{o}_{R_t}o_t\mathbf{o}_{R_t}}{\tilde{\mathcal{X}}} = \underset{\mathbf{o}_{L_t}\mathbf{o}_{R_t}}{\mathcal{M}^{-1}} \times_{\mathbf{o}_{L_t}} \underset{\mathbf{o}_{L_t}\mathbf{o}_{R_t}o_t}{\mathcal{M}} , \tag{13}$$

where the right hand side can now be estimated directly from data, without the need for the model parameters.

4.2.3 COMPUTATION OF TENSOR $\underset{o_t o_t}{\tilde{\mathcal{O}}}$

Finally, we consider the tensor

$$\underset{o_t o_t}{\tilde{\mathcal{O}}} = \underset{o_t|x_t}{\mathcal{F}^{-1}} \times_{x_t} \underset{o_t|x_t}{\mathcal{O}} . \tag{14}$$

The conditional independence relationship can take the form

$$\underset{o_t o_{t+1}}{\mathcal{M}} = \underset{o_t|x_t}{\mathcal{F}} \times_{x_t} \underset{o_{t+1}|x_t}{\mathcal{F}} \times_{x_t} \underset{x_t}{\mathcal{K}} .$$

Expressing the inverse of $\underset{o_t|x_t}{\mathcal{F}}$

$$\underset{o_t|x_t}{\mathcal{F}^{-1}} = \underset{o_t o_{t+1}}{\mathcal{M}^{-1}} \times_{o_{t+1}} \underset{o_{t+1}|x_t}{\mathcal{F}} \times_{x_t} \underset{x_t}{\mathcal{K}} ,$$

and substituting in (14), we get

$$\begin{aligned}
\underset{o_t o_t}{\tilde{\mathcal{O}}} &= \underset{o_t o_{t+1}}{\mathcal{M}^{-1}} \times_{o_{t+1}} \underset{o_{t+1}|x_t}{\mathcal{F}} \times_{x_t} \underset{x_t}{\mathcal{K}} \times_{x_t} \underset{o_t|x_t}{\mathcal{O}} \\
&= \underset{o_t o_{t+1}}{\mathcal{M}^{-1}} \times_{o_{t+1}} \underset{o_t o_{t+1}}{\mathcal{M}} .
\end{aligned} \tag{15}$$

4.3 Basic Version of Spectral Algorithm

The basic version of the spectral HSMM algorithm to compute $\underset{o_1,\ldots,o_T}{\mathcal{P}}$ entirely using the observed variables can be described as a two step process: in the learning step, compute tensors $\underset{\mathbf{o}_{R_{t-1}}\mathbf{o}_{R_t}}{\tilde{\mathcal{D}}}$, $\underset{\mathbf{o}_{R_{t-1}}o_t\mathbf{o}_{R_t}}{\tilde{\mathcal{X}}}$, and $\underset{o_t o_t}{\tilde{\mathcal{O}}}$ for each t using (10), (13) and (15) from the training data. In the inference step, use (5) to compute $p(\mathbf{S}^{test})$. Algorithm 1 shows its basic version and Figure 4 shows the graphical representation of this algorithm in terms of the transformed junction tree of Figure 2.

As an example, consider the learning step of the algorithm and the computation of tensor in (10), i.e.,

$$\underset{\mathbf{o}_{R_{t-1}}\mathbf{o}_{R_t}}{\tilde{\mathcal{D}}} = \underset{\mathbf{o}_{L_{t-1}}\mathbf{o}_{R_{t-1}}}{\mathcal{M}^{-1}} \times_{\mathbf{o}_{L_{t-1}}} \underset{\mathbf{o}_{L_{t-1}}\mathbf{o}_{R_t}}{\mathcal{M}} .$$

Algorithm 1 Basic Spectral Algorithm for HSMM inference

```
**Input:** Training sequences: $\mathbf{S}^i = \{o^i_1, \dots, o^i_{T_i}\}, i = 1, \dots, N$.
Testing sequence: $\mathbf{S}^{test} = \{o^{test}_1, \dots, o^{test}_T\}$.
**Output:** $p(\mathbf{S}^{test})$

**Learning phase:**
**for all** $t$ **do**
    Estimate $\underset{\mathbf{O}_{R_{t-1}}\mathbf{O}_{R_t}}{\tilde{\mathcal{D}}}$, $\underset{\mathbf{O}_{R_t}o_t\mathbf{O}_{R_t}}{\tilde{\mathcal{X}}}$ and $\underset{o_to_t}{\tilde{\mathcal{O}}}$ from data $\{\mathbf{S}^1, \dots, \mathbf{S}^N\}$ using equations (10), (13) and (15).
**end for**

**Inference phase:**
$p(\mathbf{S}^{test}) = 1$
**for** $t = T$ **down to** $t = 1$ **do**
    $p(\mathbf{S}^{test}) = p(\mathbf{S}^{test}) \times \underset{\mathbf{O}_{R_{t-1}}\mathbf{O}_{R_t}}{\tilde{\mathcal{D}}} \times_{\mathbf{O}_{R_t}} \left( \underset{\mathbf{O}_{R_t}o_t\mathbf{O}_{R_t}}{\tilde{\mathcal{X}}} \times_{o_t} \underset{o_to_t}{\tilde{\mathcal{O}}}\Big|_{o_t=o^{test}_t} \right)$
**end for**
```

For a fixed t, we estimate each entry of $\underset{\mathbf{O}_{L_{t-1}}\mathbf{O}_{R_{t-1}}}{\mathcal{M}}$ from the frequency of co-occurrence of tuples of the observed symbols $\{\dots, o_{t-3}, o_{t-2}, o_{t+1}, o_{t+2}, \dots\}$ in the given data set (the sets $\mathbf{O}_{L_{t-1}}$ and $\mathbf{O}_{R_{t-1}}$ were defined at the beginning of Section 4.2). Next, following our discussion after (9), we invert $\underset{\mathbf{O}_{L_{t-1}}\mathbf{O}_{R_{t-1}}}{\mathcal{M}^{-1}}$ along the modes $\mathbf{O}_{L_{t-1}}$. For this, we matrisize the tensor so that the modes $\mathbf{O}_{L_{t-1}}$ are associated with columns and $\mathbf{O}_{R_{t-1}}$ with rows in matrix $\underset{\mathbf{O}_{R_{t-1}}\mathbf{O}_{L_{t-1}}}{\mathbf{M}}$ (see Section 2 for the discussion on tensor matrisization and inversion). Finally, we compute the right inverse of the matrix to obtain $\underset{\mathbf{O}_{R_{t-1}}\mathbf{O}_{L_{t-1}}}{\mathbf{M}^{-1}}$, so that $\underset{\mathbf{O}_{R_{t-1}}\mathbf{O}_{L_{t-1}}}{\mathbf{M}} \cdot \underset{\mathbf{O}_{R_{t-1}}\mathbf{O}_{L_{t-1}}}{\mathbf{M}^{-1}} = \mathbf{I}$

Similarly, we estimate the tensor $\underset{\mathbf{O}_{L_{t-1}}\mathbf{O}_{R_t}}{\mathcal{M}}$ using the corresponding co-occurrences of the observed symbols. Matrisizing the result, so that the rows correspond to the modes $\mathbf{O}_{L_{t-1}}$ and the columns to $\mathbf{O}_{R_t}$, we get the matrix $\underset{\mathbf{O}_{L_{t-1}}\mathbf{O}_{R_t}}{\mathbf{M}}$. The multiplication $\underset{\mathbf{O}_{R_{t-1}}\mathbf{O}_{L_{t-1}}}{\mathbf{M}^{-1}} \cdot \underset{\mathbf{O}_{L_{t-1}}\mathbf{O}_{R_t}}{\mathbf{M}} = \underset{\mathbf{O}_{R_{t-1}}\mathbf{O}_{R_t}}{\tilde{\mathbf{D}}}$ produces a matrix, which is then converted to a tensor to get the final result in (10).

In the inference step we perform tensor multiplications for each t running along the length of the testing sequence. The only nuance here is that before multiplying the tensor $\underset{o_to_t}{\tilde{\mathcal{O}}}$ with others, the second mode o_t, whose dimension is n_o is collapsed into a scalar. This operation is denoted as $\underset{o_to_t}{\tilde{\mathcal{O}}}\Big|_{o_t=o^{test}_t}$, which means that based on the value of the tth symbol in testing sequence, we select the column corresponding to the element o^{test}_t. For example, if $\underset{o_to_t}{\tilde{\mathcal{O}}} \in \mathbb{R}^{10\times10}$ and $o^{test}_t = 3$ then $\underset{o_to_t}{\tilde{\mathcal{O}}}\Big|_{o_t=o^{test}_t} \in \mathbb{R}^{10\times1}$, a third column in the original matrix.

Analyzing (10), (13) and (15), we see that the computational complexity of the learning phase of the algorithm is determined by the tensor inverses and multiplications. For example, if in (10) we denote $|\mathbf{O}_R| = |\mathbf{O}_L| = \ell$ (in Section 5 we will show that $\ell = \lceil 1 + \frac{\log n_d}{\log n_x} \rceil$), then $\underset{\mathbf{O}_{L_{t-1}}\mathbf{O}_{R_{t-1}}}{\mathbf{M}} \in \mathbb{R}^{n^\ell_o \times n^\ell_o}$

and $\underset{\mathbf{O}_{L_{t-1}}\mathbf{O}_{R_t}}{\mathbf{M}} \in \mathbb{R}^{n_o^\ell \times n_o^\ell}$. The computational complexity of the multiplications and inversions would then be $\mathcal{O}(n_o^{3\ell})$. Performing this computations for all t and assuming that the length of the sequences is T, would result in $\mathcal{O}\left(n_o^{3\ell}T\right)$. Additionally, with N training examples there will be a cost of $\mathcal{O}\left(\ell NT\right)$ to estimate the sample moments $\mathfrak{M}$, which is based on counting the co-occurrences of certain observable symbols. In the inference phase of the algorithm, we perform a series of tensor multiplications with the cost of $\mathcal{O}(n_o^{3\ell}T)$.

4.4 Efficient Version of Spectral Algorithm

Note that for large ℓ the accurate estimation of tensors $\mathfrak{M}$ for each t will require large number of training sequences which might not be available, leading to inaccurate and unstable computations. Observe, however, that for example the estimated sample-based tensors $\underset{\mathbf{O}_{L_{t-1}}\mathbf{O}_{R_t}}{\mathfrak{M}}$ in (10) for each t estimate the same population quantity due to homogeneity of HSMM. Thus, a novel aspect of our work is the improvement of the accuracy and efficiency of the basic Algorithm 1 by exploiting the homogeneity property of HSMM and estimating the tensors $\tilde{\mathfrak{X}}$, $\tilde{\mathfrak{D}}$ and $\tilde{\mathfrak{O}}$ using all time steps, i.e., by pooling samples across different t and averaging the estimates. Thus, we compute only three tensors across all t, as opposed to computing these tensors separately for each t.

We show the details for computing the tensors $\tilde{\mathfrak{D}}$ in the batch form. The derivations for other tensors $\tilde{\mathfrak{X}}$ and $\tilde{\mathfrak{O}}$ can be computed in a similar manner. Recall from (10) the form of $\underset{\mathbf{O}_{R_{t-1}}\mathbf{O}_{R_t}}{\tilde{\mathfrak{D}}}$, and consider the following alternative expression, based on the sum over all t:

$$\tilde{\mathfrak{D}} = \left(\sum_t \underset{\mathbf{O}_{L_{t-1}}\mathbf{O}_{R_{t-1}}}{\mathfrak{M}}\right)^{-1} \times_{\mathbf{O}_L} \left(\sum_t \underset{\mathbf{O}_{L_{t-1}}\mathbf{O}_{R_t}}{\mathfrak{M}}\right), \tag{16}$$

where $\mathbf{O}_L$ denotes a generic mode of the averaged tensor $\mathfrak{M}$, corresponding to $\mathbf{O}_{L_{t-1}}$ for all t. Note that in practice, instead of summation, we use averaging to avoid numerical overflow problems, and the average is equivalent to the expression in (16) since the term $\frac{1}{T}$ cancels out. Since

$$\underset{\mathbf{O}_{L_{t-1}}\mathbf{O}_{R_{t-1}}}{\mathfrak{M}} = \underset{\mathbf{O}_{L_{t-1}}|x_{t-1}d_{t-2}}{\mathfrak{F}} \times_{x_{t-1}d_{t-2}} \underset{\mathbf{O}_{R_{t-1}}|x_{t-1}d_{t-2}}{\mathfrak{F}} \times_{x_{t-1}d_{t-2}} \underset{x_{t-1}d_{t-2}}{\mathfrak{K}}, \tag{17}$$

the first term inside brackets can be rewritten as:

$$\begin{aligned} \sum_t \underset{\mathbf{O}_{L_{t-1}}|x_{t-1}d_{t-2}}{\mathfrak{F}} \times_{x_{t-1}d_{t-2}} \underset{\mathbf{O}_{R_{t-1}}|x_{t-1}d_{t-2}}{\mathfrak{F}} \times_{x_{t-1}d_{t-2}} \underset{x_{t-1}d_{t-2}}{\mathfrak{K}} &\overset{(a)}{=} \sum_t \underset{\mathbf{O}_{R_{t-1}}|x_{t-1}d_{t-2}}{\mathfrak{F}} \times_{x_{t-1}d_{t-2}} \underset{\mathbf{O}_{L_{t-1}}x_{t-1}d_{t-2}}{\overline{\mathfrak{F}}} \\ &\overset{(b)}{=} \underset{\mathbf{O}_{R_2}|x_2d_1}{\mathfrak{F}} \times \left(\sum_t \underset{\mathbf{O}_{L_{t-1}}x_{t-1}d_{t-2}}{\overline{\mathfrak{F}}}\right), \end{aligned} \tag{18}$$

where in (a) we combined the two factors, i.e., $\underset{\mathbf{O}_{L_{t-1}}x_{t-1}d_{t-2}}{\overline{\mathfrak{F}}} = \underset{\mathbf{O}_{L_{t-1}}|x_{t-1}d_{t-2}}{\mathfrak{F}} \times_{x_{t-1}d_{t-2}} \underset{x_{t-1}d_{t-2}x_{t-1}d_{t-2}}{\mathfrak{K}}$ and in (b) we used the homogeneity property of HSMM, i.e., the fact that $\underset{\mathbf{O}_{R_{t-1}}|x_{t-1}d_{t-2}}{\mathfrak{F}}$ does not depend on time stamp t, and extracted one of the common factors, in fact, the first factor. Note that the term $\underset{\mathbf{O}_{L_{t-1}}x_{t-1}d_{t-2}}{\overline{\mathfrak{F}}}$, on the other hand, does depend on t since the factor $\underset{x_{t-1}d_{t-2}}{\mathfrak{K}}$, which represents the probability $p(x_{t-1}, d_{t-2})$, changes as the time stamp t changes.

Similarly, since

$$\underset{\mathbf{O}_{L_{t-1}}\mathbf{O}_{R_t}}{\mathcal{M}} = \underset{\mathbf{O}_{L_{t-1}}|x_{t-1}d_{t-2}}{\mathcal{F}} \times_{x_{t-1}d_{t-2}} \underset{x_{t-1}d_{t-2}}{\mathcal{K}} \times_{x_{t-1}d_{t-2}} \underset{d_{t-1}|x_{t-1}x_{t-1}d_{t-2}}{\mathcal{D}} \times_{x_{t-1}d_{t-1}} \underset{\mathbf{O}_{R_t}|x_{t-1}d_{t-1}}{\mathcal{F}}, \tag{19}$$

rewrite the second term in (16) as

$$\begin{aligned}
&\sum_t \underset{\mathbf{O}_{L_{t-1}}|x_{t-1}d_{t-2}}{\mathcal{F}} \times_{x_{t-1}d_{t-2}} \underset{x_{t-1}d_{t-2}}{\mathcal{K}} \times_{x_{t-1}d_{t-2}} \underset{d_{t-1}|x_{t-1}x_{t-1}d_{t-2}}{\mathcal{D}} \times_{x_{t-1}d_{t-1}} \underset{\mathbf{O}_{R_t}|x_{t-1}d_{t-1}}{\mathcal{F}} \\
&\quad = \sum_t \underset{\mathbf{O}_{L_{t-1}}x_{t-1}d_{t-2}}{\overline{\mathcal{F}}} \times_{x_{t-1}d_{t-2}} \underset{d_{t-1}|x_{t-1}x_{t-1}d_{t-2}}{\mathcal{D}} \times_{x_{t-1}d_{t-1}} \underset{\mathbf{O}_{R_t}|x_{t-1}d_{t-1}}{\mathcal{F}} \\
&\quad = \left(\sum_t \underset{\mathbf{O}_{L_{t-1}}x_{t-1}d_{t-2}}{\overline{\mathcal{F}}}\right) \times \underset{d_2|x_2x_2d_1}{\mathcal{D}} \times_{x_2d_2} \underset{\mathbf{O}_{R_3}|x_2d_2}{\mathcal{F}},
\end{aligned} \tag{20}$$

where we used the transformations similar as in (18), i.e., the fact that the factors $\underset{d_{t-1}|x_{t-1}x_{t-1}d_{t-2}}{\mathcal{D}}$ and $\underset{\mathbf{O}_{R_t}|x_{t-1}d_{t-1}}{\mathcal{F}}$ are homogeneous, independent of t. Now if we multiply the inverse of (18) with (20), we get

$$\underset{\mathbf{O}_{R_2}|x_2d_1}{\mathcal{F}^{-1}} \times \left(\sum_t \underset{\mathbf{O}_{L_{t-1}}x_{t-1}d_{t-2}}{\overline{\mathcal{F}}}\right)^{-1} \times \left(\sum_t \underset{\mathbf{O}_{L_{t-1}}x_{t-1}d_{t-2}}{\overline{\mathcal{F}}}\right) \times \underset{d_2|x_2x_2d_1}{\mathcal{D}} \times \underset{\mathbf{O}_{R_3}|x_2d_2}{\mathcal{F}} \tag{21}$$

$$\begin{aligned}
&= \underset{\mathbf{O}_{R_2}|x_2d_1}{\mathcal{F}^{-1}} \times_{x_2d_1} \underset{d_2|x_2x_2d_1}{\mathcal{D}} \times_{x_2d_2} \underset{\mathbf{O}_{R_3}|x_2d_2}{\mathcal{F}} \\
&= \underset{\mathbf{O}_{R_2}\mathbf{O}_{R_3}}{\tilde{\mathcal{D}}} = \underset{\mathbf{O}_{R_{t-1}}\mathbf{O}_{R_t}}{\tilde{\mathcal{D}}},
\end{aligned} \tag{22}$$

where in (21) we used the fact that the order in which tensors are multiplied is irrelevant and also the fact that the terms in parenthesis are invertible. This is due to the fact that the set of observations $\mathbf{O}_{L_{t-1}}$ for all t is selected so as to make each of the summand invertible (see Section 5 for the details about the choice of $\mathbf{O}_{L_{t-1}}$). Moreover, in (22) we used the definition of $\underset{\mathbf{O}_{R_{t-1}}\mathbf{O}_{R_t}}{\tilde{\mathcal{D}}}$

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

together with the homogeneity property of HSMM. We note that although the above derivations rely on the assumption of the existence of the matrix summation inverse in equation (21), the idea of aggregating observations from multiple time steps has also been utilized by other works, e.g., (Siddiqi et al., 2010; Anandkumar et al., 2014a) and shown to be very effective in practice, significantly improving the accuracy of corresponding algorithms.

We can conclude that the batch form of the tensor takes the form:

$$\tilde{\mathcal{D}} = \left(\sum_t \underset{\mathbf{O}_{L_{t-1}}\mathbf{O}_{R_{t-1}}}{\mathcal{M}}\right)^{-1} \times_{\mathbf{O}_L} \left(\sum_t \underset{\mathbf{O}_{L_{t-1}}\mathbf{O}_{R_t}}{\mathcal{M}}\right). \tag{23}$$

Algorithm 2 Efficient Spectral Algorithm for HSMM inference

```
**Input:** Training sequences: $\mathbf{S}^i = \{o_1^i, \ldots, o_{T_i}^i\}, i = 1, \ldots, N$.
Testing sequence: $\mathbf{S}^{test} = \{o_1^{test}, \ldots, o_T^{test}\}$.
**Output:** $p(\mathbf{S}^{test})$

**Learning phase:**
Estimate $\tilde{\mathcal{D}}, \tilde{\mathcal{X}}$ and $\tilde{\mathcal{O}}$ from data $\{\mathbf{S}^1, \ldots, \mathbf{S}^N\}$ using equations (23), (24) and (25).

**Inference phase:**
$p(\mathbf{S}^{test}) = 1$
**for** $i = T$ **down to** $i = 1$ **do**
    $p(\mathbf{S}^{test}) = p(\mathbf{S}^{test}) \times \tilde{\mathcal{D}} \times \left(\tilde{\mathcal{X}} \times \tilde{\mathcal{O}}|_{o=o_i^{test}}\right)$
**end for**
```

Similar derivations can be carried out to obtain the rest of the tensors in the batch form:

$$\tilde{\mathcal{X}} = \left(\sum_t \underset{\mathbf{O}_{L_t}\mathbf{O}_{R_t}}{\mathcal{M}}\right)^{-1} \times_{\mathbf{O}_L} \left(\sum_t \underset{\mathbf{O}_{L_t}\mathbf{O}_{R_t}o_t}{\mathcal{M}}\right) , \tag{24}$$

$$\tilde{\mathcal{O}} = \left(\sum_t \underset{o_t o_{t+1}}{\mathcal{M}}\right)^{-1} \times_o \left(\sum_t \underset{o_t o_{t+1}}{\mathcal{M}}\right) . \tag{25}$$

where in the last expression the mode o corresponds to the mode $o_{t_{t+1}}$ after averaging of tensor $\underset{o_t o_{t+1}}{\mathcal{M}}$ for all t.

Analyzing (23), (24) and (25), we see that the computational complexity of the learning phase of the Algorithm 2 is now $\mathcal{O}\left((n_o^{2\ell} + \ell N)T\right)$, mainly determined by the tensor additions and the estimation of the sample moments $\mathcal{M}$. The number of inverses and multiplications is now fixed and independent of sequence length T. Specifically, there will be only three tensor multiplications and inversions for a total cost of $\mathcal{O}(n_o^{3\ell})$ (as opposed to T tensor multiplications and inversions as in Algorithm 1). The computational complexity of the inference phase is $\mathcal{O}(n_o^{3\ell}T)$, which is the same as for Algorithm 1.

Note that such a batch tensor computation significantly improves the accuracy of the resulting spectral algorithm. In part, this is due to the fact that we now use more data to estimate the tensors as compared to the original form (5). The estimates obtained in this form have lower variance, which in turn ensures that the inverses we compute in (23), (24) and (25) are more stable and accurate.

5. Rank Analysis of Observable Tensors

In Section 4.2.1, when we derived the equations (10), (13) and (15), 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 $\underset{o_t o_{t+1}}{\mathcal{M}^{-1}}$. 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 (10) and expand all its terms using (8) to get

$$\underset{\mathsf{O}_{R_{t-1}}\mathsf{O}_{R_t}}{\tilde{\mathfrak{D}}} = \underset{\mathsf{O}_{R_{t-1}}|x_{t-1}d_{t-2}}{\mathfrak{F}^{-1}} \times \overbracket{\underset{\mathsf{O}_{L_{t-1}}|x_{t-1}d_{t-2}}{\mathfrak{F}^{-1}} \times \underbracket{\underset{x_{t-1}d_{t-2}}{\mathfrak{K}^{-1}} \times \underset{x_{t-1}d_{t-2}}{\mathfrak{K}}} \times \underset{\mathsf{O}_{L_{t-1}}|x_{t-1}d_{t-2}}{\mathfrak{F}}} \times \underset{d_{t-1}|x_{t-1}x_{t-1}d_{t-2}}{\mathfrak{D}} \times \underset{\mathsf{O}_{R_t}|x_{t-1}d_{t-1}}{\mathfrak{F}},$$

where we dropped the multiplication subscripts and some of the duplicated modes, which can be inferred from the context. Observe that in order for the above equation to produce (6), the terms in the middle must multiply out into identity tensor

$$\underset{x_{t-1}d_{t-2}}{\mathfrak{I}} = \underset{x_{t-1}d_{t-2}}{\mathfrak{K}^{-1}} \times_{x_{t-1}d_{t-2}} \underset{x_{t-1}d_{t-2}}{\mathfrak{K}} \qquad \underset{x_{t-1}d_{t-2}}{\mathfrak{I}} = \underset{\mathsf{O}_{L_{t-1}}|x_{t-1}d_{t-2}}{\mathfrak{F}^{-1}} \times_{\mathsf{O}_{L_{t-1}}} \underset{\mathsf{O}_{L_{t-1}}|x_{t-1}d_{t-2}}{\mathfrak{F}}. \tag{26}$$

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

$$\underset{x_{t-1}d_{t-2}}{\mathfrak{I}} = \underset{\mathsf{O}_{R_{t-1}}|x_{t-1}d_{t-2}}{\mathfrak{F}^{-1}} \times_{\mathsf{O}_{R_{t-1}}} \underset{\mathsf{O}_{R_{t-1}}|x_{t-1}d_{t-2}}{\mathfrak{F}}, \tag{27}$$

therefore, we can conclude that for (10) to exist, the identity statements in (26) and (27) must be satisfied. These statements have implications for the ranks of $\underset{x_{t-1}d_{t-2}}{\mathfrak{K}}$, $\underset{\mathsf{O}_{L_{t-1}}|x_{t-1}d_{t-2}}{\mathfrak{F}}$ and $\underset{\mathsf{O}_{R_{t-1}}|x_{t-1}d_{t-2}}{\mathfrak{F}}$, which in turn determine the length of the observation sequences $\mathsf{O}_{L_{t-1}}$ and $\mathsf{O}_{R_{t-1}}$.

Since $\underset{x_{t-1}d_{t-2}}{\mathfrak{K}}$ 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 assumptions $A1$ and $A2$, 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 the first equation in (26) is satisfied.

Next, consider the second equation in (26) and recall from Section 2 that if we matrisize the tensor as $\underset{\mathsf{O}_{L_{t-1}}|x_{t-1}d_{t-2}}{\mathbf{F}} \in \mathbb{R}^{n_o^{|\mathsf{O}_{L_{t-1}}|} \times n_x n_d}$ then $\mathbf{F}$ must have full column rank $n_x n_d$ for the proper inverse to exist, implying $n_o^{|\mathsf{O}_{L_{t-1}}|} \geq n_x n_d$. Similarly, $\underset{\mathsf{O}_{R_{t-1}}|x_{t-1}d_{t-2}}{\mathfrak{F}}$ in (27) must have rank $n_x n_d$. As a consequence of the above, the tensor

$$\underset{\mathsf{O}_{L_{t-1}}\mathsf{O}_{R_{t-1}}}{\mathfrak{M}} = \underset{\mathsf{O}_{L_{t-1}}|x_{t-1}d_{t-2}}{\mathfrak{F}} \times \underset{\mathsf{O}_{R_{t-1}}|x_{t-1}d_{t-2}}{\mathfrak{F}} \times \underset{x_{t-1}d_{t-2}}{\mathfrak{K}} \tag{28}$$

will have rank $n_x n_d$ and, in general, is rank-deficient.

The argument above can also be used to show that $\underset{\mathsf{O}_{L_t}\mathsf{O}_{R_t}}{\mathfrak{M}}$ has rank $n_x n_d$ since in (12) the tensors $\underset{x_{t-1}d_{t-1}}{\mathfrak{K}}$, $\underset{\mathsf{O}_{L_t}|x_{t-1}d_{t-1}}{\mathfrak{F}}$ and $\underset{\mathsf{O}_{R_t}|x_{t-1}d_{t-1}}{\mathfrak{F}}$ all have rank $n_x n_d$. Similarly, $\underset{o_t o_{t+1}}{\mathfrak{M}}$ will have rank n_x because in (15) the rank of the participating tensors $\underset{x_t}{\mathfrak{K}}$, $\underset{o_{t+1}|x_t}{\mathfrak{F}}$ and $\underset{o_t|x_t}{\mathfrak{F}}$ is n_x. In particular, note that the tensor $\underset{o_t|x_t}{\mathfrak{F}}$ is the observation matrix $O \in \mathbb{R}^{n_o \times n_x}$ of the model and it has rank n_x according to assumption $A3$. This conclusion also justifies our choice for $\omega_{x_t} = o_t$ at the end of Section 4.1.

The key unknowns now are the sets of the observed variables O_R and O_L that must be appropriately selected for the corresponding tensors to have rank $n_x n_d$. Recall that we defined $\mathsf{O}_{R_{t-1}} = \{o_t, o_{t+1}, \ldots\}$. As one of the new key results 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:

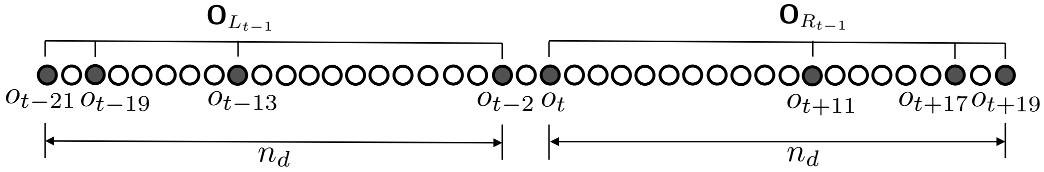

Figure 5: Observations required to estimate $\underset{\mathbf{O}_{L_{t-1}}\mathbf{O}_{R_{t-1}}}{\mathcal{M}}$ from data for HSMM with $n_x = 3$ and $n_d = 20$.

Theorem 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 $n_x^\ell \geq n_x n_d$ and we want the minimal ℓ which satisfies this. 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 $\mathbf{O}_{R_{t-1}} = \{o_t, o_{t+11}, o_{t+17}, o_{t+19}\}$ and $\mathbf{O}_{L_{t-1}} = \{o_{t-21}, o_{t-19}, o_{t-13}, o_{t-2}\}$, where the set $\mathbf{O}_{L_{t-1}}$ is defined similar to $\mathbf{O}_{R_{t-1}}$ in Theorem 1 but for the indices to the left of time stamp $t-1$. Figure 5 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.

In order to prove the above Theorem, we will focus our analysis on the tensor $\underset{\mathbf{O}_{R_{t+1}}|x_t d_t}{\mathcal{F}}$ instead of $\underset{\mathbf{O}_{R_{t-1}}|x_{t-1}d_{t-2}}{\mathcal{F}}$. This specific choice was only done to ensure the compactness in our notations, however the HSMM homogeneity property enables us to transfer this result for tensors for any t. Note that

$$\underset{\mathbf{O}_{R_{t+1}}|x_t d_t}{\mathcal{F}} = \underset{\mathbf{O}_{R_{t-1}}|x_{t-2}d_{t-2}}{\mathcal{F}} = \underset{\mathbf{O}_{R_{t-1}}|x_{t-1}d_{t-2}}{\mathcal{F}} \times_{x_{t-1}d_{t-2}} \underset{x_{t-1}d_{t-2}|x_{t-2}d_{t-2}}{\mathcal{X}},$$

where the first equality is due to the homogeneity property of the model and in the second equality we embedded the HSMM transition matrix into tensor $\underset{x_{t-1}d_{t-2}|x_{t-2}d_{t-2}}{\mathcal{X}}$ with mode d_{t-2} duplicated. It can be shown that the matricized tensor $\underset{x_{t-1}d_{t-2}|x_{t-2}d_{t-2}}{\mathbf{X}} \in \mathbb{R}^{n_x n_d \times n_x n_d}$ has rank $n_x n_d$, i.e., it is full rank. Therefore, the rank structure of $\underset{\mathbf{O}_{R_{t+1}}|x_t d_t}{\mathcal{F}}$ determines the rank structure of $\underset{\mathbf{O}_{R_{t-1}}|x_{t-1}d_{t-2}}{\mathcal{F}}$.

The rest of Section 5 is devoted to the proof of Theorem 1. We first establish the rank structure of tensor $\underset{\mathbf{O}_{R_{t+1}}|x_t d_t}{\mathcal{F}}$ for sequential set of observations $\mathbf{O}_{R_{t+1}}$ and then analyze the rank structure for the observations which were selected non-sequentially.

5.1 Rank Structure of Tensor $\underset{\mathbf{O}_{R_{t+1}}|x_t d_t}{\mathcal{F}}$

Define by $\mathbf{X}_{R_{t+1}} = \{x_{t+2}, x_{t+3}, \ldots\}$, the sequence of hidden states corresponding to observations $\mathbf{O}_{R_{t+1}} = \{o_{t+2}, o_{t+3}, \ldots\}$. Then using conditional independence property of the graphical model in Figure 1, namely, that the variables $\mathbf{O}_{R_{t+1}}$ and $x_t d_t$ are independent given $\mathbf{X}_{R_{t+1}}$, we can write:

$$\underset{\mathbf{O}_{R_{t+1}}|x_t d_t}{\mathcal{F}} = \underset{\mathbf{O}_{R_{t+1}}|\mathbf{X}_{R_{t+1}}}{\mathcal{Q}} \times \underset{\mathbf{X}_{R_{t+1}}|x_t d_t}{\mathcal{T}}, \tag{29}$$

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 (29) can be written as $\mathbf{Q} = \otimes_\ell O \in \mathbb{R}^{n_o^\ell \times n_x^\ell}$, i.e., a Kronecker product of the observation matrix O with itself ℓ times. According to the assumption $A3$, $rank(O) = n_x$ 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 (29) 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 invertibility of $\mathcal{F}$, we need to select a set of variables $\mathbf{X}_{R_{t+1}}$ so that $rank\left(\underset{\mathbf{X}_{R_{t+1}}|x_t d_t}{\mathbf{T}}\right) = n_x n_d$ with the condition that $n_x^\ell \geq n_x n_d$. Thus, the problem of the analysis of the rank structure of tensor $\underset{\mathbf{O}_{R_{t+1}}|x_t d_t}{\mathcal{F}}$ translates to the problem of rank structure of matrix $\underset{\mathbf{X}_{R_{t+1}}|x_t d_t}{\mathbf{T}}$. In what follows, we assume that $\mathbf{X}_{R_{t+1}} = \{x_{t+2}, \ldots, x_{t+\ell+1}\}$ are sequential and so we would be interested in determining ℓ which makes $rank\left(\underset{\mathbf{X}_{R_{t+1}}|x_t d_t}{\mathbf{T}}\right) = n_x n_d$. Later, the sequential assumption will be removed and we show how to select such variables in a more efficient way.

5.1.1 COMPUTATION OF FACTOR T

In order to study the rank structure of $\underset{\mathbf{X}_{R_{t+1}}|x_t d_t}{\mathbf{T}}$ we will have to understand the mechanism how this matrix is constructed and how the rank changes as the size of $\mathbf{X}_{R_{t+1}}$ increases. We start by considering the following conditional independence relationships from the model in Figure 1:

$$p(x_{t+3}, x_{t+2}|x_{t+1}, d_{t+1}) = \sum_{d_{t+2}} p(x_{t+3}|x_{t+2}, d_{t+2})\, \underline{p(d_{t+2}|x_{t+2}, d_{t+1}) p(x_{t+2}|x_{t+1}, d_{t+1})} \tag{30}$$

$$p(x_{t+3}, x_{t+2}, x_{t+1}|x_t, d_t) = \sum_{d_{t+1}} p(x_{t+3}, x_{t+2}|x_{t+1}, d_{t+1})\, \underline{p(d_{t+1}|x_{t+1}, d_t) p(x_{t+1}|x_t, d_t)}\,. \tag{31}$$

Using the model's homogeneity property, we see that the quantity underlined in expression (30) is the same as the one in (31). Moreover, equation (30) can then be thought of as transforming $p(x_{t+1}|x_t, d_t)$ into $p(x_{t+2}, x_{t+1}|x_t, d_t)$, while the expression in (31), in effect, transforms probability $p(x_{t+2}, x_{t+1}|x_t, d_t)$ into $p(x_{t+3}, x_{t+2}, x_{t+1}|x_t, d_t)$. Thus (30) and (31) encode the following chain of transformations:

$$p(x_{t+1}|x_t, d_t) \to p(x_{t+2}, x_{t+1}|x_t, d_t) \to p(x_{t+3}, x_{t+2}, x_{t+1}|x_t, d_t).$$

Based on the above considerations, we can rewrite (30) and (31) in the tensor form as follows:

$$\underset{x_{t+3},x_{t+2}|x_{t+1},d_{t+1}}{\mathcal{T}} = \underset{x_{t+3},x_{t+2}|x_{t+2},d_{t+2}}{\mathcal{T}} \times_{x_{t+2}d_{t+2}} \underset{x_{t+2},d_{t+2}|x_{t+1}d_{t+1}}{\mathcal{V}} \tag{32}$$

$$\underset{x_{t+3},x_{t+2},x_{t+1}|x_t,d_t}{\mathcal{T}} = \underset{x_{t+3},x_{t+2},x_{t+1}|x_{t+1},d_{t+1}}{\mathcal{T}} \times_{x_{t+1}d_{d+1}} \underset{x_{t+1},d_{t+1}|x_t d_t}{\mathcal{V}}, \tag{33}$$

where $\underset{x_{t+2},d_{t+2}|x_{t+1},d_{t+1}}{\mathcal{V}} = \underset{x_{t+1},d_{t+1}|x_t,d_t}{\mathcal{V}} = \underset{x_{t+1},d_{t+1}|x_{t+1},d_t}{\mathcal{D}} \times_{x_{t+1}d_t} \underset{x_{t+1},d_t|x_t,d_t}{\mathcal{X}}$. The homogeneity property allows us to rewrite the above as

$$\underset{x_{t+2},x_{t+1}|x_t,d_t}{\mathcal{T}} = \underset{x_{t+1},x_t|x_t,d_t}{\mathcal{T}} \times \mathcal{V} \tag{34}$$

$$\underset{x_{t+3},x_{t+2},x_{t+1},x_{t+1}|x_t,d_t}{\mathcal{T}} = \underset{x_{t+2},x_{t+1}|x_t,d_t}{\mathcal{T}} \times \mathcal{V}. \tag{35}$$

Our next step is to represent the above tensor equations in the matrix form. First, consider tensor $\mathcal{V}$, its matricized form can be written as:

$$\mathbf{V} = \underset{x_{t+1},d_{t+1}|x_{t+1},d_t}{\mathbf{D}} \; \underset{x_{t+1},d_t|x_t,d_t}{\mathbf{X}} \tag{36}$$

where $\underset{x_{t+1},d_{t+1}|x_{t+1},d_t}{\mathbf{D}} \in \mathbb{R}^{n_x n_d \times n_x n_d}$ and $\underset{x_{t+1},d_t|x_t,d_t}{\mathbf{X}} \in \mathbb{R}^{n_x n_d \times n_x n_d}$. Next, consider the equations (34) and (35), its matrix version is of the form:

$$\underset{x_{t+2},x_{t+1}|x_t,d_t}{\mathbf{T}} = \underset{x_{t+1},x_t|x_t,d_t}{\mathbf{T}} \; \mathbf{V} \tag{37}$$

$$\underset{x_{t+3},x_{t+2},x_{t+1}|x_t,d_t}{\mathbf{T}} = \underset{x_{t+2},x_{t+1},x_t|x_t,d_t}{\mathbf{T}} \; \mathbf{V}, \tag{38}$$

here the sizes of the matrices are $\underset{x_{t+1},x_t|x_t,d_t}{\mathbf{T}} \in \mathbb{R}^{n_x^2 \times n_x n_d}$, $\underset{x_{t+2},x_{t+1}|x_t,d_t}{\mathbf{T}} \in \mathbb{R}^{n_x^2 \times n_x n_d}$, and similarly $\underset{x_{t+2},x_{t+1},x_t|x_t,d_t}{\mathbf{T}} \in \mathbb{R}^{n_x^3 \times n_x n_d}$, and matrix $\underset{x_{t+3},x_{t+2},x_t|x_t,d_t}{\mathbf{T}} \in \mathbb{R}^{n_x^3 \times n_x n_d}$.

Summarizing the above derivations, we can describe the following algorithmic approach for analyzing $\underset{\mathsf{X}_{R_{t+1}}|x_t d_t}{\mathbf{T}}$ as $\mathsf{X}_{R_{t+1}}$ increases. We begin with $\underset{x_{t+1}|x_t,d_t}{\mathbf{T}} = [\mathcal{X}\, \mathbf{I} \cdots \mathbf{I}] \in \mathbb{R}^{n_x \times n_x n_d}$, where the first block $\mathcal{X} \in \mathbb{R}^{n_x \times n_x}$ corresponds to $d_t = 1$, and the subsequent $(n_d - 1)$ blocks of $\mathbf{I} \in \mathbb{R}^{n_x \times n_x}$ correspond to $d_t > 1$ for which $x_{t+1} = x_t$. We then use (37) to get $\underset{x_{t+2},x_{t+1}|x_t,d_t}{\mathbf{T}}$. However, notice that in (37) the matrix $\underset{x_{t+1},x_t|x_t,d_t}{\mathbf{T}}$ has a duplicated mode x_t, therefore, we need to restructure $\underset{x_{t+1}|x_t,d_t}{\mathbf{T}}$, which can be accomplished with:

$$\underset{x_{t+1},x_t|x_t,d_t}{\mathbf{T}'} = \underset{x_{t+1}|x_t,d_t}{\mathbf{T}} \odot \mathbf{E},$$

where $\mathbf{E} = [\mathbf{I} \cdots \mathbf{I}] \in \mathbb{R}^{n_x \times n_x n_d}$ and $\odot$ denotes a Khatri-Rao product (row-wise Kronecker product)[2]. Then, we use (38) to transform $\underset{x_{t+2},x_{t+1}|x_t,d_t}{\mathbf{T}}$ into $\underset{x_{t+3},x_{t+2},x_{t+1}|x_t,d_t}{\mathbf{T}}$ where, again a

[2] Let $\mathbf{P} = \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \vdots \\ \mathbf{p}_n \end{bmatrix} \in \mathbb{R}^{m \times n}$ and $\mathbf{Q} \in \mathbb{R}^{k \times n}$ then $\mathbf{P} \odot \mathbf{Q} = \begin{bmatrix} \mathbf{p}_1 \otimes \mathbf{Q} \\ \mathbf{p}_2 \otimes \mathbf{Q} \\ \vdots \\ \mathbf{p}_n \otimes \mathbf{Q} \end{bmatrix} \in \mathbb{R}^{mk \times n}$, where $\otimes$ is a Kronecker product.

```
**Algorithm 3** Computation of $\underset{\mathbf{X}_{R_{t+1}}|x_t d_t}{\mathbf{T}}$
  **Input:** $p(d_t|x_t, d_{t-1})$ and $p(x_t|x_{t-1}, d_{t-1})$ - duration and transition distributions, $\ell$ - the number of sequential hidden states represented by $\mathbf{X}_{R_{t+1}}$.
  **Initialization:**
      $p(x_{t+1}|x_t, d_t) \rightarrow \underset{x_{t+1}|x_t,d_t}{\mathbf{T}}$
      $p(d_{t+1}|x_{t+1}, d_t) \rightarrow \underset{x_{t+1},d_{t+1}|x_{t+1},d_t}{\mathbf{D}}$
      $p(x_{t+1}|x_t, d_t) \rightarrow \underset{x_{t+1},d_t|x_t,d_t}{\mathbf{X}}$
      $\mathbf{V} = \underset{x_{t+1},d_{t+1}|x_{t+1},d_t}{\mathbf{D}} \underset{x_{t+1},d_t|x_t,d_t}{\mathbf{X}}, \quad \mathbf{E} = [\mathbf{I} \cdots \mathbf{I}]$
  **for** $i = 1$ **to** $\ell - 1$ **do**
      $\underset{x_{t+i},\,\dots\,,x_{t+1},x_t|x_t,d_t}{\mathbf{T}'} = \underset{x_{t+i},\,\dots\,,x_{t+1}|x_t,d_t}{\mathbf{T}} \odot \mathbf{E}$     (39)
      $\underset{x_{t+i+1},\,\dots\,,x_{t+2},x_{t+1}|x_t,d_t}{\mathbf{T}} = \underset{x_{t+i},\,\dots\,,x_{t+1},x_t|x_t,d_t}{\mathbf{T}'} \mathbf{V}$     (40)
  **end for**
```

preliminary step is to restructure the matrix as follows:

$$\underset{x_{t+2},x_{t+1},x_t|x_t,d_t}{\mathbf{T}'} = \underset{x_{t+2},x_{t+1}|x_t,d_t}{\mathbf{T}} \odot \mathbf{E}.$$

Algorithm 3 summarizes the above constructions for a general case.

The following Theorem characterizes the rank structure of matrix $\underset{\mathbf{X}_{R_{t+1}}|x_t d_t}{\mathbf{T}}$ in the output of the Algorithm 3. The proof can be found in Appendix A.1.

Theorem 2 *The rank of the output matrix* $\underset{\mathbf{X}_{R_{t+1}}|x_t d_t}{\mathbf{T}}$ *in Algorithm 3 is* $\min(\ell n_x, n_x n_d)$.

Applying now Theorem 2 to equation (29) in matrix form

$$\underset{\mathbf{O}_{R_{t+1}}|x_t d_t}{\mathbf{F}} = \underset{\mathbf{O}_{R_{t+1}}|\mathbf{X}_{R_{t+1}}}{\mathbf{Q}} \times \underset{\mathbf{X}_{R_{t+1}}|x_t d_t}{\mathbf{T}},$$

where $rank(\mathbf{Q}) = n_x^\ell$ we can now conclude the following result:

Corollary 3 *To achieve the full column rank for* $\underset{\mathbf{O}_{R_{t+1}}|x_t d_t}{\mathbf{F}} \in \mathbb{R}^{n_o^\ell \times n_x n_d}$, *i.e. to ensure that the rank of tensor* $\underset{\mathbf{O}_{R_{t+1}}|x_t d_t}{\mathcal{F}}$ *is* $n_x n_d$, *the number of observations* ℓ *in* $\mathbf{O}_{R_{t+1}} = \{o_{t+2}, o_{t+3}, \dots, o_{t+\ell+1}\}$ *must be equal to the maximum state persistence i.e.,* $\ell = n_d$.

Algorithm 4 Efficient computation of $\underset{\mathsf{X}_{R_{t+1}}|x_t d_t}{\mathbf{T}}$

```
Input: $p(d_t|x_t, d_{t-1})$ and $p(x_t|x_{t-1}, d_{t-1})$ - duration and transition distributions, $\ell$ - the number of sequential hidden states represented by $\mathsf{X}_{R_{t+1}}$
Initialization:
    $p(x_{t+1}|x_t, d_t) \rightarrow \underset{x_{t+1}|x_t,d_t}{\mathbf{T}}$
    $p(d_{t+1}|x_{t+1}, d_t) \rightarrow \underset{x_{t+1},d_{t+1}|x_{t+1},d_t}{\mathbf{D}}$
    $p(x_{t+1}|x_t, d_t) \rightarrow \underset{x_{t+1},d_t|x_t,d_t}{\mathbf{X}}$
    $\mathbf{V} = \underset{x_{t+1},d_{t+1}|x_{t+1},d_t}{\mathbf{D}} \; \underset{x_{t+1},d_t|x_t,d_t}{\mathbf{X}}, \quad \mathbf{E} = [\mathbf{I} \cdots \mathbf{I}]$
$c = 1$
for $i = 1$ to $\ell - 1$ do
        $\mathbf{T} = \mathbf{T} \;\; \mathbf{V}$        (41)
    if $i == (n_x)^c - 1$ or $i == \ell - 1$ do
        $\mathbf{T} = \mathbf{T} \odot \; \mathbf{E}$        (42)
    end if
    $c = c + 1$
end for
```

5.1.2 EFFICIENT COMPUTATION OF FACTOR $\mathbf{T}$

In Corollary 3 we established that the number of observations in $\mathsf{O}_{R_{t+1}} = \{o_{t+2}, o_{t+3}, \ldots, o_{t+\ell+1}\}$ is $\ell = n_d$. Therefore, the sizes of the estimated quantities $\tilde{\mathfrak{D}} \in \mathbb{R}^{n_o^{n_d} \times n_o^{n_d}}$ and $\tilde{\mathfrak{X}} \in \mathbb{R}^{n_o^{n_d} \times n_o^{n_d} \times n_o}$ in Algorithm 3 will have exponential dependency on n_d. When maximum state persistence is large, the estimation of such quantity becomes impractical. Fortunately, we can modify Algorithm 3 to significantly reduce the number of observations. The idea is to apply the step (40) multiple times in-between the applications of step (39). Recall that in the previous construction we established that $\ell = n_d$ consecutive observations are sufficient, e.g., $\mathsf{O}_{R_{t+1}} = \{o_{t+2}, \ldots, o_{t+\ell+1}\}$. In contrast, in the proposed approach, every time we add an observation, say $o_{t+\tau}$, we skip certain number δ of time steps before adding another observation $o_{t+\tau+\delta}$, so that the observations are non-consecutive. As we illustrate next, the span of these non-consecutive observations is still n_d but the number of them is only logarithmic in n_d. The proposed approach still achieves the full rank structure of $\underset{\mathsf{O}_{R_{t+1}}|x_t d_t}{\mathbf{F}}$ but with smaller number of data points. Algorithm 4, which is a simple modification of Algorithm 3, summarizes the above procedure.

The following result establishes the rank structure of the matrix $\underset{\mathsf{X}_{R_{t+1}}|x_td_t}{\mathbf{T}}$ in the output of the Algorithm 4. The proof can be found in Appendix A.2.

Theorem 4 *The rank of the output matrix* $\underset{\mathsf{X}_{R_{t+1}}|x_td_t}{\mathbf{T}}$ *in Algorithm 4 is* $\min(n_x^\ell, n_x n_d)$.

Note that based on the above theorem, Algorithm 4 increases the rank at every step exponentially rather than linearly. In order for $\underset{\mathsf{X}_{R_{t+1}}|x_td_t}{\mathbf{T}}$ to achieve the rank $n_x n_d$ we will now require $\ell = \lceil 1 + \frac{\log n_d}{\log n_x} \rceil$ observations, since we need to ensure $n_x^\ell = n_x n_d$. Observe that the span of the selected observations is still n_d, while the number of the observations is only logarithmic in n_d. The following Corollary summarizes the above conclusions.

Corollary 5 *To achieve the full column rank for* $\underset{\mathsf{O}_{R_{t+1}}|x_td_t}{\mathbf{F}} \in \mathbb{R}^{n_o^\ell \times n_x n_d}$*, i.e. to ensure that the rank of tensor* $\underset{\mathsf{O}_{R_{t+1}}|x_td_t}{\mathfrak{F}}$ *is* $n_x n_d$*, the number of observations* ℓ *in* $\mathsf{O}_{R_{t+1}}$ *must be equal to* $\ell = \lceil 1 + \frac{\log n_d}{\log n_x} \rceil$*, since we need to ensure* $n_x^\ell = n_x n_d$.

Theorem 4 together with Corollary 5 now proves the Theorem 1 stated earlier.

6. Experiments

In this section we evaluated the performance of the proposed algorithm both on synthetic as well as real data sets 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 proposed 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 $N_{train} = \{500, 1000, 5000, 10^4, 10^5\}$ training and $N_{test} == 1000$ 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 1000 such values, we then computed the final score, which is the root-mean-square error (RMSE) across all the testing sequences, RMSE $= \sqrt{\frac{1}{N_{test}} \sum_{i=1}^{N_{test}} \epsilon_i^2}$.

Figure 6 shows results, where the top row of graphs corresponds to the model with $n_o = 3, n_x = 2, n_d = 2$ and the bottom row is for model with $n_o = 5, n_x = 4, n_d = 6$. The left column of graphs shows the progression of RMSE across EM iterations for both models; the middle column shows the dependence of testing error on the number of training samples and the right column shows the running times. It can be observed from plots (b) and (e) in Figure 6 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, i.e., whose n_o, n_x and n_d are larger, 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 (f) in Figure 6 show that spectral method is several orders of magnitude faster than EM.

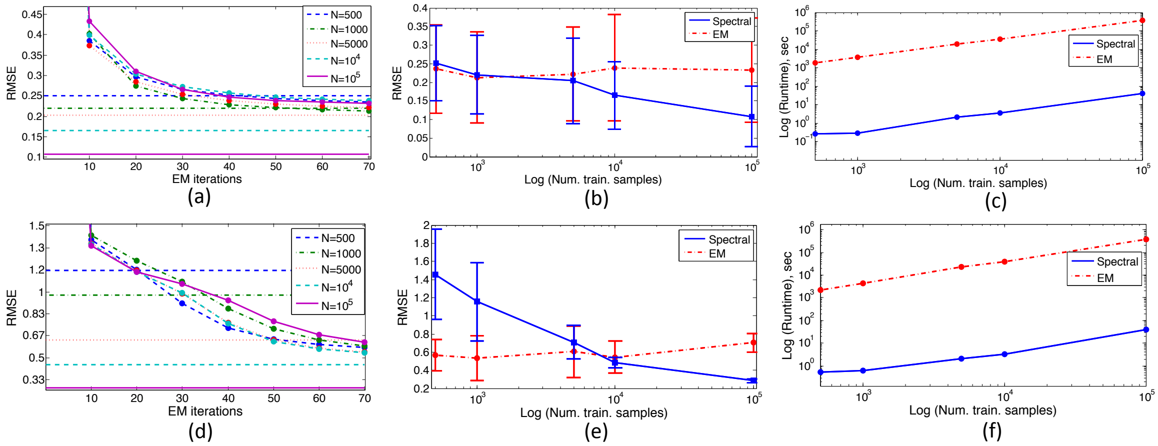

Figure 6: Performance of the spectral algorithm and EM on synthetic data generated from HSMM with $n_o = 3, n_x = 2, n_d = 2$ (top row) and $n_o = 5, n_x = 4, n_d = 6$ (bottom row). (a), (d): Error for EM across different iterations for various training data sets. The straight lines show the performance 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.

Given the above results, we can conclude that (i) for small data sets EM is a preferable algorithm, (ii) for large data, the spectral algorithm is a better choice, since it achieves higher accuracy and (iii) across all data sets the spectral algorithm requires significantly less computations as compared to EM.

6.2 Application to Aviation Safety Data

We also compared the performance of the spectral algorithm and EM on real NASA flight data set (NASA), containing over $180,000$ flights of 35 aircrafts from a defunct 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 (Budalakoti et al., 2009; Gorinevsky et al., 2012; Matthews et al., 2013) and used HSMM to detect abnormal flights based on pilot actions. Our idea is based on the observation that a flight can be partitioned into a number of phases, e.g., initial descent, touch down, or braking on the runway, and where within each phase the pilot performs certain actions. For example, during the initial descent, the pilot reduces throttle, lowers the flaps, and uses the ailerons and elevator to stabilize the aircraft. On the other hand, in the braking stage, the pilot uses brakes as well as rudder to keep the aircraft in the middle of the runway. Using HSMM as a model, we represented the flight phases as hidden states and the pilot actions as the observations from these states (see Melnyk et al. (2013) and Melnyk et al. (2016) for more details).

In our experiments, we focused on a part of flight related to the landing phase, which typically lasts 15-60 minutes in duration from when the flight crosses 10,000 ft while approaching an airport

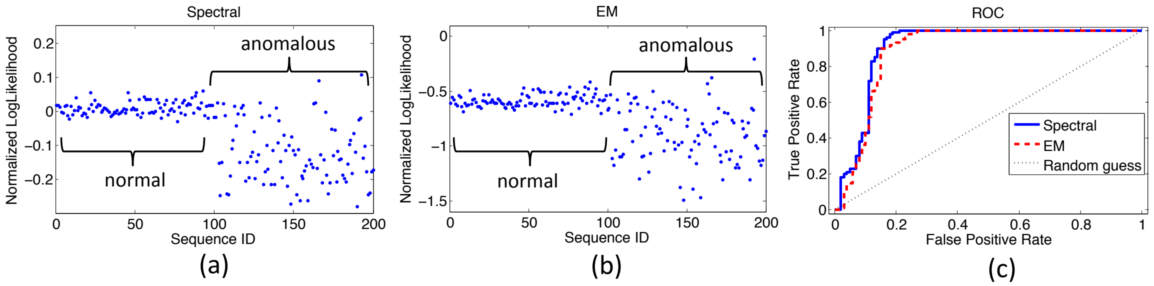

Figure 7: Evaluation of the spectral algorithm and EM on aviation safety data. (a) and (b): Normalized joint loglikelihood computed by spectral algorithm (a) and EM (b) for a set of 200 test flights, with 100 normal and 100 anomalous. HSMM parameters: $n_o = 9, n_x = 8, n_d = 40$ (c): The Receiver Operating Characteristic (ROC) curve, illustrating classification accuracy of the algorithms. Area Under Curve (AUC) for spectral algorithm is 0.91 and for EM is 0.89.

to the touch down on the runway. Our experiments are run on a subset of flights landing at the same airport. We chose 9 pilot commands, which include "selected altitude", "selected heading", "selected throttle level", etc. A simple data filter, based on the histogram of the pilot actions, was applied to select $10,020$ normal flights for training. The 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 anomalous flights contained low occurrence or rare events, such as fast descent, unusual usage of air brakes, etc., and few significant anomalies, e.g., aborted descent in order to delay the flight when the runway is not available. The length of the considered flight sequences varied from 500 to 4000 seconds.

For each flight in the testing set, we applied EM and spectral algorithm to compute the normalized joint log-likelihood

$$\frac{1}{T_i} \log p(o_1, o_2, \ldots, o_{T_i}),$$

where o_i are the observed pilot actions for test flight i, and T_i is the length of the test flight i, with $i = 1, \ldots, 200$. Figure 7 shows the results. The high-likelihood sequences were considered normal and low-likelihood ones classified as anomalous (see plots (a) and (b) in Figure 7). 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 $= 0.89$ (see plot (c) on Figure 7). On the other hand, the computational time of the spectral algorithm was orders of magnitude smaller as compared to EM. We also compared performance of both algorithm on the same flight data while varying the dimensionality of the HSMM parameters (see Figure 8 and Table 2). We can see that although the performance of EM and spectral algorithm is similar across many models, the latter offers significant computational savings.

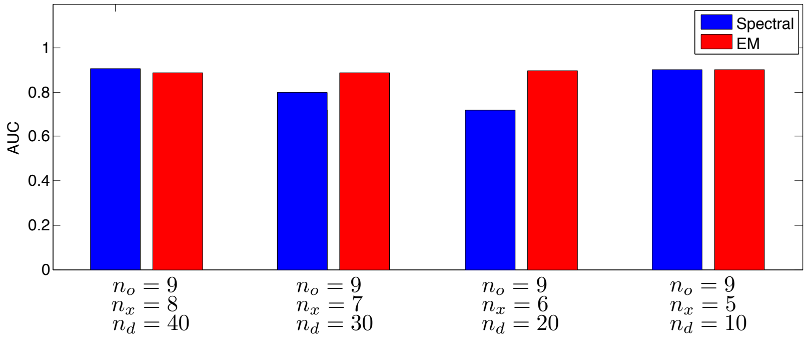

Figure 8: Comparison of AUC scores for EM and spectral algorithm for various model parameters when evaluated on aviation safety data. Both algorithms achieve similar high accuracy across different models.

Parameters		$n_o = 9$ $n_x = 8$ $n_d = 40$	$n_o = 9$ $n_x = 7$ $n_d = 30$	$n_o = 9$ $n_x = 6$ $n_d = 20$	$n_o = 9$ $n_x = 5$ $n_d = 10$
Running Time	Spectral	6.8 hours	6.4 hours	6.4 hours	6.3 hours
	EM	> 2 days	> 2 days	> 2 days	> 2 days

Table 2: Comparison of running time for EM and spectral algorithm for multiple model parameters on the flight data. Spectral algorithm is several orders of magnitude faster as compared to EM, offering significant computational savings.

7. Conclusion

In this paper, we presented a novel spectral algorithm to perform inference in HSMM. We derived an observable representation of the model which can be computed from the data sample moments of size logarithmic in the maximum length of latent state persistence. Based on the representation and exploiting the homogeneity of the model, we presented an efficient approach to inference, which ensures that during the training phase the number of matrix multiplications and inverses is fixed and independent of the sequence length of the observations. Empirical evaluation on synthetic and real flight data sets were presented to illustrate the promise of the proposed spectral algorithm. In particular, the spectral method gets similar or better performance than EM as the size of the training data set increases, and at the same time the spectral method is orders of magnitude faster than EM providing significant computational savings. Going forward, we plan to explore if similar spectral methods can be developed for inference in more general dynamic Bayesian networks.

Acknowledgments

We thank Nikunj Oza and Bryan Matthews at NASA for their helpful comments and suggestions at several stages of the work. We thank the editor and reviewers for helpful comments and suggestions which led to improvements in the paper. We also thank the Minnesota Supercomputing Institute

(MSI) for the computing support. This work was supported by NASA grant NNX12AQ39A, and by NSF grants IIS-1563950, IIS-1447566, IIS-1447574, IIS-1422557, CCF-1451986, CNS- 1314560, IIS-0953274, IIS-1029711, and gifts from Adobe, IBM, and Yahoo.

Appendix A. Analysis of Tensor Rank Structure

A.1 Analysis of Algorithm 3

In this Section, we provide analysis of the Algorithm 3 and study the rank structure of matrix $\mathbf{T}$ in order to prove Theorem 2. To understand the analysis, it is important to know how the structure of matrix $\mathbf{T}_{\mathbf{X}_{R_{t+1}}|x_t d_t}$ evolves across iterations. For this, we present in Figure 9 a schematic description of a few steps of the algorithm. For the analysis we will need to establish certain auxiliary results.

Lemma 6 *Let $\mathbf{A} \in \mathbb{R}^{m\times n}$ be a matrix which has no all-zero columns, then the $rank\,(\mathbf{I} \odot \mathbf{A}) = rank\,(\mathbf{A} \odot \mathbf{I}) = n$, where $\odot$ denotes Khatri-Rao product and $\mathbf{I} \in \mathbb{R}^{n\times n}$.*

Proof Let $\mathbf{K} = (\mathbf{I} \odot \mathbf{A}) \in \mathbb{R}^{mn\times n}$. By definition of Khatri-Rao product, $\mathbf{K}(:, j) = \mathbf{e}_j \otimes \mathbf{A}(:, j)$, for $j = 1, \dots, n$, which consists of zeros, except for rows $(j-1)m+1, \dots, (j-1)m+m$, containing the column $\mathbf{A}(:, j)$. Here $\otimes$ denotes Kronecker product and $\mathbf{e}_j$ is everywhere zero except for

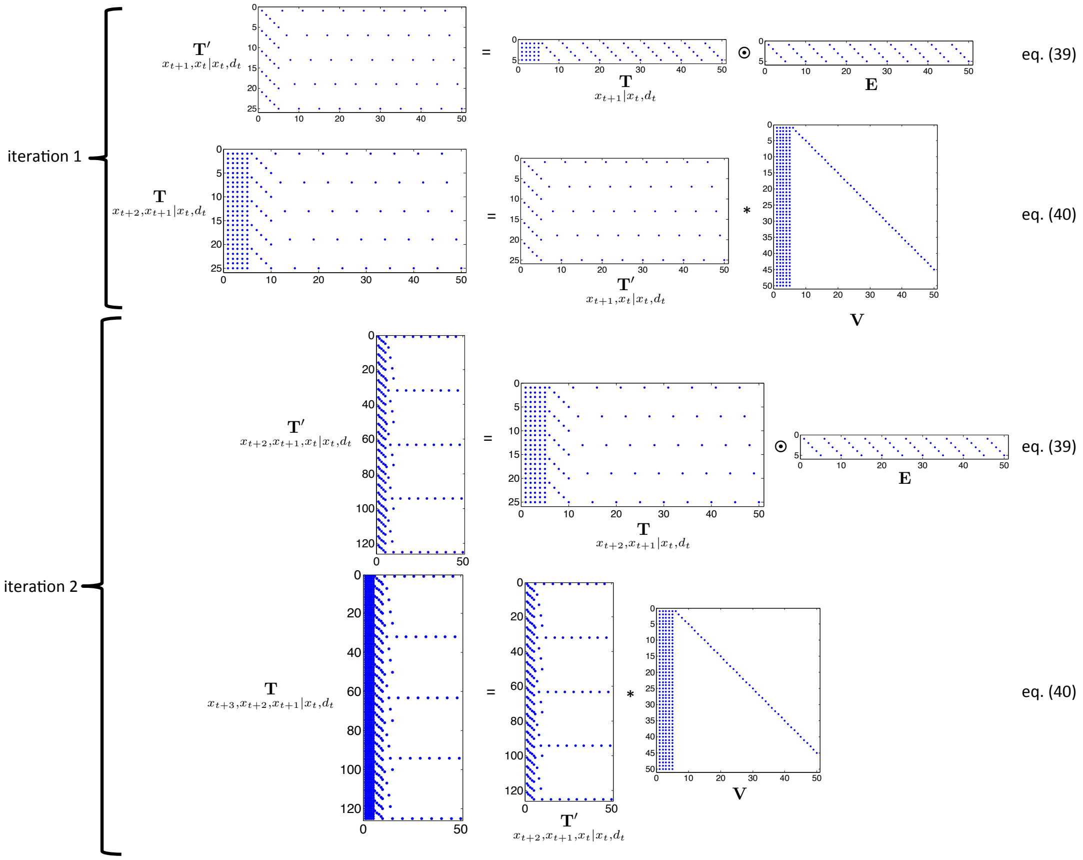
iteration 1
$\mathbf{T}'_{x_{t+1},x_t|x_t,d_t}$
=
$\mathbf{T}_{x_{t+1}|x_t,d_t}$
$\odot$
$\mathbf{E}$
eq. (39)
$\mathbf{T}_{x_{t+2},x_{t+1}|x_t,d_t}$
=
$\mathbf{T}'_{x_{t+1},x_t|x_t,d_t}$
*
$\mathbf{V}$
eq. (40)
iteration 2
$\mathbf{T}'_{x_{t+2},x_{t+1},x_t|x_t,d_t}$
=
$\mathbf{T}_{x_{t+2},x_{t+1}|x_t,d_t}$
$\odot$
$\mathbf{E}$
eq. (39)
$\mathbf{T}_{x_{t+3},x_{t+2},x_{t+1}|x_t,d_t}$
=
$\mathbf{T}'_{x_{t+2},x_{t+1},x_t|x_t,d_t}$
*
$\mathbf{V}$
eq. (40)
0
5
10
15
20
25
30
35
40
45
50
60
80
100
120

Figure 9: Schematic representation of Algorithm 3. This example illustrates the HSMM with $n_x = 5$ and $n_d = 10$. The non-zero matrix elements are displayed as dots.

position j which is 1. As long as there is no all-zero columns in $\mathbf{A}$, each column of $\mathbf{K}$ is independent of each other and therefore the rank is n. Moreover, since the matrix $\mathbf{A} \odot \mathbf{I}$ is a row-permuted version of $\mathbf{A} \odot \mathbf{I}$, their ranks are the same. ■

Lemma 7 *Define a block-row matrix* $\mathbf{M} = [\mathbf{A}_1\ \mathbf{A}_2\ \cdots\ \mathbf{A}_k] \in \mathbb{R}^{m \times kn}$*, where each* $\mathbf{A}_i \in \mathbb{R}^{m \times n}$*. Define by* $r_j,\ j = 1, \dots, n$ *the rank of matrix* $[\mathbf{A}_1(:,j)\ \cdots\ \mathbf{A}_k(:,j)]$ *composed of jth columns of* $\mathbf{A}$*'s, and let* $\mathbf{E} = [\mathbf{I}\,\mathbf{I}\ \cdots\ \mathbf{I}] \in \mathbb{R}^{n \times kn}$*, where* $\mathbf{I} \in \mathbb{R}^{n \times n}$*. Then the rank of matrix* $\mathbf{W} = \mathbf{M} \odot \mathbf{E} \in \mathbb{R}^{mn \times kn}$*, obtained using a Khatri-Rao product, is* $\min(mn, \sum_j r_j)$*.*

Proof First note that $\mathbf{M} \odot \mathbf{E}$ and $\mathbf{E} \odot \mathbf{M}$ are row permuted version of each other, so they have the same rank. Therefore, consider $\mathbf{W}' = \mathbf{E} \odot \mathbf{M} = [\mathbf{I} \odot \mathbf{A}_1 \cdots \mathbf{I} \odot \mathbf{A}_k]$. Also, note that $\mathbf{e}_j \otimes [\mathbf{A}_1(:,j) \cdots\ \mathbf{A}_k(:,j)]$, $j = 1, \dots, n$ is a matrix which consists of zeros except for rows $(j-1)m + 1, \dots, (j-1)m + m$ where it contains the columns $[\mathbf{A}_1(:,j)\ \cdots\ \mathbf{A}_k(:,j)]$. The rank of these columns is r_j and all other columns in $\mathbf{W}$ are independent of them due to the structure of the Khatri-Rao product. Therefore, each set of such columns adds r_j to the total rank. Since the overall rank of $\mathbf{W}$ cannot exceed either the number of rows or columns, we conclude that $rank(\mathbf{W}) = \min(mn, \sum_j r_j)$. ■

Lemma 8 *Let* $V = \{\mathbf{v}_1, \dots, \mathbf{v}_n\}$ *be a set of linearly independent vectors. Define* $\mathbf{u} = \sum_{i=1}^{n} c_i \mathbf{v}_i$*, where coefficients* $c_i \neq 0, i = 1, \dots, n$*. Define* U *to be a strict subset of* V*, i.e.,* $U \subset V$*, then a set of vectors* $\mathbf{u} \cup U$ *is independent.*

Proof Define $\{1, \dots, n\} = \alpha \cup \bar{\alpha}$, where α denotes a subset of indices for vectors corresponding to U. Then we can write $\mathbf{u} = \sum_{i:i\in\alpha} c_i \mathbf{v}_i + \sum_{j:j\in\bar{\alpha}} c_j \mathbf{v}_j$.

Assuming the opposite, i.e., $\mathbf{u} \cup U$ are dependent, we can write $k_0 \mathbf{u} + \sum_{i:i\in\alpha} k_i \mathbf{v}_i = 0$ where $k_0 \neq 0$ and some of $k_i, i \in \alpha$ are also must be non-zero. Substituting the definition of $\mathbf{u}$ and rearranging the terms, we get:

$$k_0 \sum_{i:i\in\alpha} (c_i + k_i)\mathbf{v}_i + k_0 \sum_{j:j\in\bar{\alpha}} c_j \mathbf{v}_j = 0.$$

Since $c_j \neq 0, j \in \bar{\alpha}$, the above equation claims the linear dependence of vectors in V, which is a contradiction of our assumption and so $\mathbf{u} \cup U$ are independent. ■

We are now ready to analyze Algorithm 3. It can be verified that (36) is of the form:

$$\mathbf{V} = \left[\begin{array}{c|ccc} & \mathbf{I} & & \\ \Psi & & \ddots & \\ & & & \mathbf{I} \\ \hline & \mathbf{0} & \cdots & \mathbf{0} \end{array}\right] \in \mathbb{R}^{n_x n_d \,\times\, n_x n_d} \qquad \text{where} \quad \Psi = \begin{bmatrix} \text{diag}\,[D(1,:)]\,\mathcal{X} \\ \text{diag}\,[D(2,:)]\,\mathcal{X} \\ \vdots \\ \text{diag}\,[D(n_d,:)]\,\mathcal{X} \end{bmatrix} \in \mathbb{R}^{n_x n_d \,\times\, n_x}, \tag{43}$$

where diag $[D(i,:)]$ is the diagonal matrix with ith row from D on the diagonal. Note that we can also write $\Psi = (D \odot \mathbf{I})\,\mathcal{X}$. Observe that the rank of $\mathbf{V}$ is $n_x n_d$ because the $n_x(n_d - 1) \times n_x(n_d - 1)$

block diagonal matrix delineated in (43) and the last $n_x \times n_x$ block matrix diag $[D(n_d, :)]\,\mathcal{X}$ in Ψ together comprising $n_x n_d$ independent columns of $\mathbf{V}$. Note that diag $[D(n_d, :)]\,\mathcal{X}$ has rank n_x because $\mathcal{X}$ is full rank and $D(n_d, :)$ is non-zero, which follows from assumptions $A1$ and $A2$. As a side note observe that the requirement to have $D(n_d, :)$ non-zero implies that there is a non-zero probability of maximum state persistence.

In analyzing the Algorithm 3, it would be useful to denote the matrices at iteration i in (39) and (40) as

$$\begin{aligned}\underset{x_{t+i},\,\dots\,,x_{t+1}|x_t,d_t}{\mathbf{T}} &= [\mathbf{A}_1^{(i)}\ \cdots\ \mathbf{A}_{n_d}^{(i)}]\\ \underset{x_{t+i},\,\dots\,,x_{t+1},x_t|x_t,d_t}{\mathbf{T}'} &= [\mathbf{B}_1^{(i)}\ \cdots\ \mathbf{B}_{n_d}^{(i)}]\\ \underset{x_{t+i+1},\dots,x_{t+2},x_{t+1}|x_t,d_t}{\mathbf{T}} &= [\mathbf{C}_1^{(i)}\ \cdots\ \mathbf{C}_{n_d}^{(i)}].\end{aligned}$$

Moreover, utilizing the structure of matrix $\mathbf{V}$ from (43), the operations involved in step (40) are as follows:

$$\Big[\mathbf{C}_1^{(i)}\ \mathbf{C}_2^{(i)}\ \mathbf{C}_3^{(i)}\ \ \cdots\ \mathbf{C}_{n_d}^{(i)}\Big] = \Big[[\mathbf{B}_1^{(i)}\ \cdots\ \mathbf{B}_{n_d}^{(i)}]\Psi\ \ \mathbf{B}_1^{(i)}\ \mathbf{B}_2^{(i)}\ \ \cdots\ \ \mathbf{B}_{n_d-1}^{(i)}\Big]. \tag{44}$$

With the above information we can now present the proof of Theorem 2:
Proof of Theorem 2 At the start of the algorithm $\underset{x_{t+1}|x_t,d_t}{\mathbf{T}} = [\mathcal{X}\ \mathbf{I}\ \cdots\ \mathbf{I}] = [\mathbf{A}_1^{(1)} \cdots \mathbf{A}_{n_d}^{(1)}]$, which has rank n_x. The rank of matrix $\Big[\mathbf{A}_1^{(1)}(:, l) \cdots \mathbf{A}_{n_d}^{(1)}(:, l)\Big]$ for $l = 1, \dots, n_x$ is $r_l = 2$ since among all the columns only two of them are independent. Therefore, according to Lemma 7, the result of operations in (39), has rank $\sum_l r_l = 2n_x$. Moreover, we note that since $[\mathbf{B}_1^{(1)}\ \mathbf{B}_2^{(1)}\ \cdots\ \mathbf{B}_{n_d}^{(1)}] = [\mathcal{X}\odot\mathbf{I}\ \mathbf{I}\odot\mathbf{I}\ \cdots\ \mathbf{I}\odot\mathbf{I}]$, it can be seen that its $2n_x$ independent vectors can be formed by the columns $[\mathbf{B}_1^{(1)}\ \mathbf{B}_2^{(1)}]$, so that the rank of $\Big[\mathbf{B}_1^{(1)}(:, l) \cdots \mathbf{B}_{n_d}^{(1)}(:, l)\Big]$ for $l = 1, \dots, n_x$ is 2.

Next, since the rank of $\mathbf{V}$ is $n_x n_d$, the operations in (40) produce matrix $[\mathbf{C}_1^{(1)}\ \mathbf{C}_2^{(1)}\ \cdots\ \mathbf{C}_{n_d}^{(1)}]$ with the rank still being $2n_x$. Moreover, the columns of $\mathbf{C}_1^{(1)}$ are linearly dependent on the rest of the columns, $[\mathbf{C}_2^{(1)}\ \cdots\ \mathbf{C}_{n_d}^{(1)}]$, due to (44). However, the rank of $\Big[\mathbf{C}_1^{(1)}(:, l) \cdots \mathbf{C}_{n_d}^{(1)}(:, l)\Big]$ is now $r_l = 3$ for $l = 1, \dots, n_x$. To understand this, note that

$$\begin{aligned}[\mathbf{B}_1^{(1)}\ \mathbf{B}_2^{(1)}\ \cdots\ \mathbf{B}_{n_d}^{1}] &= [\mathcal{X}\odot\mathbf{I}\ \mathbf{I}\odot\mathbf{I}\ \cdots\ \mathbf{I}\odot\mathbf{I}]\\ [\mathbf{C}_1^{(1)}\ \mathbf{C}_2^{(1)}\ \mathbf{C}_3^{(1)}\ \cdots\ \mathbf{C}_{n_d}^{(1)}] &= [\mathbf{C}_1^{(1)}\ \mathcal{X}\odot\mathbf{I}\ \mathbf{I}\odot\mathbf{I}\ \cdots\ \mathbf{I}\odot\mathbf{I}],\end{aligned}$$

where, according to (44), $\mathbf{C}_1^{(1)} = [\mathbf{B}_1^{(1)} \cdots \mathbf{B}_{n_d}^{(1)}]\Psi$. As we established before, the rank of the matrix $\Big[\mathbf{C}_2^{(1)}(:, l) \cdots \mathbf{C}_{n_d}^{(1)}(:, l)\Big] = \Big[\mathbf{B}_1^{(1)}(:, l) \cdots \mathbf{B}_{n_d-1}^{(1)}(:, l)\Big]$ is $r_l = 2$. Moreover, it can also be checked that $\mathbf{C}_1^{(1)}(:, l)$ is independent of $\Big[\mathbf{C}_2^{(1)}(:, l) \cdots \mathbf{C}_{n_d}^{(1)}(:, l)\Big]$ due to Lemma 8. Clearly, then the cumulative rank of $\Big[\mathbf{C}_1^{(1)}(:, l) \cdots \mathbf{C}_{n_d}^{(1)}(:, l)\Big]$ is 3 for $l = 1, \dots, n_x$.

To generalize the above, if at the iteration i the rank of matrix $\Big[\mathbf{A}_1^{(i)} \cdots \mathbf{A}_{n_d}^{(i)}\Big]$ is in_x while the rank of $\Big[\mathbf{A}_1^{(i)}(:, l) \cdots \mathbf{A}_{n_d}^{(i)}(:, l)\Big]$ is $(i + 1)$, then the operations in step (39) produce $\Big[\mathbf{B}_1^{(i)} \cdots \mathbf{B}_{n_d}^{(i)}\Big]$

having rank $(i+1)n_x$ due to Lemma 7. The step in (40) keeps the rank of $\left[\mathbf{C}_1^{(i)} \cdots \mathbf{C}_{n_d}^{(i)}\right]$ at $(i+1)n_x$ due to the full rank structure of $\mathbf{V}$. At the same time, this step increases the rank of matrix $\left[\mathbf{C}_1^{(i)}(:,l) \cdots \mathbf{C}_{n_d}^{(i)}(:,l)\right]$ to $(i+2)$ due to Lemma 8, i.e., independence of $\mathbf{C}_1^{(i)}(:,l)$ from $\left[\mathbf{C}_2^{(i)}(:,l) \cdots \mathbf{C}_{n_d}^{(i)}(:,l)\right]$ with the latter having the rank $(i+1)$. Therefore, each iteration increases the rank of matrix $\mathbf{T}$ by n_x and so after $2 \leq \ell \leq n_d$ steps the rank of the resulting matrix $\underset{\mathbf{x}_{R_{t+1}}|x_t d_t}{\mathbf{T}}$ is ℓn_x.

Note that if $\ell = 1$ then the Algorithm 3 is not executed and returns the trivial $\underset{x_{t+1}|x_t,d_t}{\mathbf{T}}$ with rank n_x. On the other hand, if $\ell > n_d$ then the rank of $\underset{\mathbf{x}_{R_{t+1}}|x_t d_t}{\mathbf{T}}$ is $n_x n_d$ since this is the number of columns in that matrix and so is the maximum achievable rank. ■

A.2 Analysis of Algorithm 4

In this Section we analysis of the Algorithm 4 in order to prove Theorem 4. Similarly as in Section A.1, it is instructive to visualize the progress of Algorithm 4. Figure 10 shows a schematic description of a few steps of the algorithm.

We are now ready to present the proof of Theorem 4.

Proof of Theorem 4 For the proof, we refer back to Algorithm 3 and the proof of Theorem 2. Recall, that at iteration $i = 1$, the result of step (39) is a matrix $[\mathbf{B}_1^{(1)} \cdots \mathbf{B}_{n_d}^{(1)}] \in \mathbb{R}^{n_x^2 \times n_x n_d}$, whose rank is $2n_x$, since $\left[\mathbf{A}_1^{(1)}(:,l) \cdots \mathbf{A}_{n_d}^{(1)}(:,l)\right] = [\mathcal{X}\,\mathbf{I} \cdots \mathbf{I}] \in \mathbb{R}^{n_x \times n_x n_d}$ for $l = 1, \ldots, n_x$ had two independent columns. Then, the transformations in step (40) produced $\left[\mathbf{C}_1^{(1)}(:,l) \cdots \mathbf{C}_{n_d}^{(1)}(:,l)\right]$ for $l = 1, \ldots, n_x$ with rank $3n_x$.

Note that if $n_x > 2$ then $\left[\mathbf{A}_1^{(1)}(:,l) \cdots \mathbf{A}_{n_d}^{(1)}(:,l)\right]$ potentially can have a rank up to n_x, while in Algorithm 3 we only have it equal to 2. It turns out that if we apply step (40) multiple times and use Lemma 8, we can increase the rank of $\left[\mathbf{C}_1^{(1)}(:,l) \cdots \mathbf{C}_{n_d}^{(1)}(:,l)\right]$ for $l = 1, \ldots, n_x$ to n_x.

Specifically, consider step (41). At iteration $i = 1$ we have $[\mathbf{A}_1^{(1)} \cdots \mathbf{A}_{n_d}^{(1)}] = [\mathbf{B}_1^{(1)} \cdots \mathbf{B}_{n_d}^{(1)}]$ and for $l = 1, \ldots, n_x$ the two independent columns are $\left[\mathbf{B}_1^{(1)}(:,l)\ \mathbf{B}_2^{(1)}(:,l)\right] = [\mathcal{X}(:,l)\ \mathbf{I}(:,l)]$. The result of step (41) gives us then three independent columns

$$\left[\mathbf{C}_1^{(1)}(:,l)\ \mathbf{C}_2^{(1)}(:,l)\ \mathbf{C}_3^{(1)}(:,l)\right] = \left[\mathbf{C}_1^{(1)}(:,l)\ \mathcal{X}(:,l)\ \mathbf{I}(:,l)\right],$$

where $\mathbf{C}_1^{(1)} = [\mathcal{X}\,\mathbf{I} \cdots \mathbf{I}]\Psi$. The independence follows from Lemma 8. The repeated application of step (41) one more time gives four independent columns

$$\left[\mathbf{C}_1^{(2)}(:,l)\ \mathbf{C}_2^{(2)}(:,l)\ \mathbf{C}_3^{(2)}(:,l)\ \mathbf{C}_4^{(2)}(:,l)\right] = \left[\mathbf{C}_1^{(2)}(:,l)\ \mathbf{C}_1^{(1)}(:,l)\ \mathcal{X}(:,l)\ \mathbf{I}(:,l)\right],$$

where $\mathbf{C}_1^{(2)} = [\mathbf{C}_1^{(1)} \cdots \mathbf{C}_{n_d}^{(1)}]\Psi$. Observe that since the number of rows is n_x, we can increase the rank at most up to n_x. Therefore, if in the beginning we had *two* independent columns and we want to get n_x independent columns, we would need to apply the step (41) $n_x - 2$ times, so as to have the matrix $[\mathbf{C}_1^{(n_x-2)}(:,l)\ \cdots\ \mathbf{C}_{n_d}^{(n_x-2)}(:,l)]$ with rank n_x.

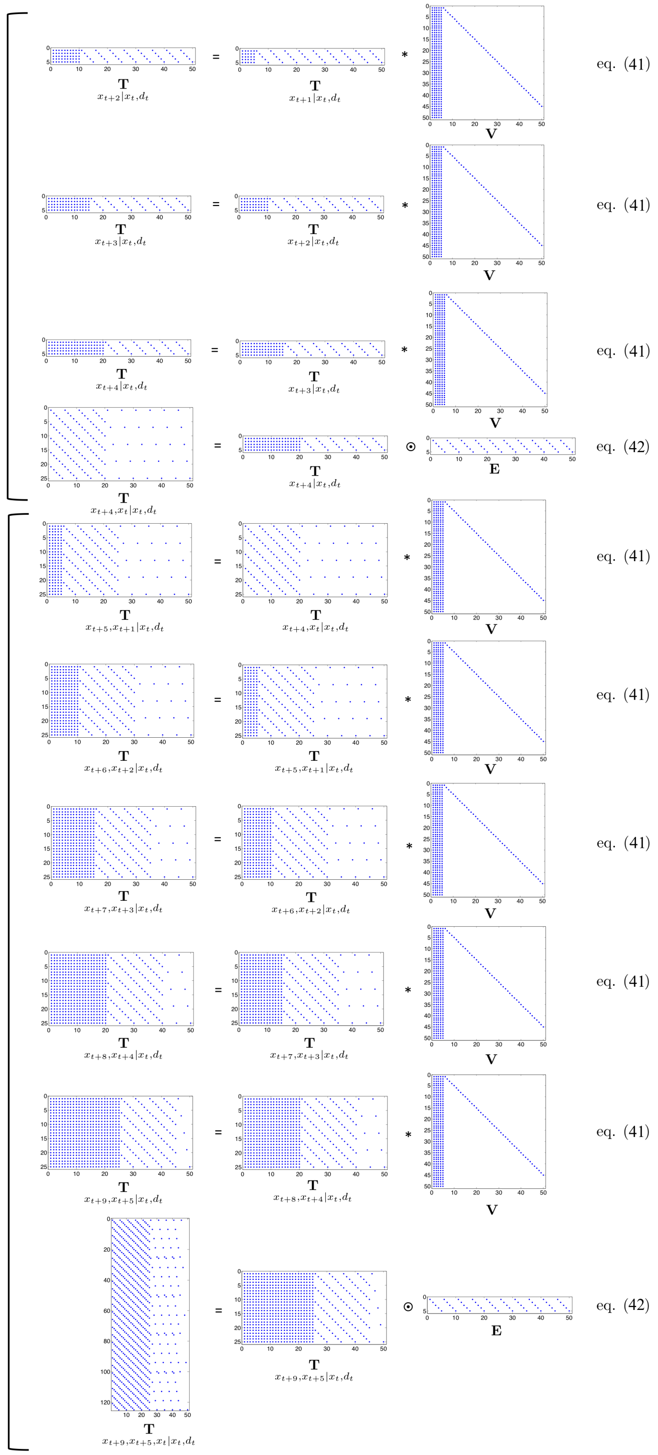
$\mathbf{T}_{x_{t+2}|x_t,d_t}$ = $\mathbf{T}_{x_{t+1}|x_t,d_t}$ * $\mathbf{V}$ eq. (41)
$\mathbf{T}_{x_{t+3}|x_t,d_t}$ = $\mathbf{T}_{x_{t+2}|x_t,d_t}$ * $\mathbf{V}$ eq. (41)
$\mathbf{T}_{x_{t+4}|x_t,d_t}$ = $\mathbf{T}_{x_{t+3}|x_t,d_t}$ * $\mathbf{V}$ eq. (41)
$\mathbf{T}_{x_{t+4},x_t|x_t,d_t}$ = $\mathbf{T}_{x_{t+4}|x_t,d_t}$ ⊙ $\mathbf{E}$ eq. (42)
$\mathbf{T}_{x_{t+5},x_{t+1}|x_t,d_t}$ = $\mathbf{T}_{x_{t+4},x_t|x_t,d_t}$ * $\mathbf{V}$ eq. (41)
$\mathbf{T}_{x_{t+6},x_{t+2}|x_t,d_t}$ = $\mathbf{T}_{x_{t+5},x_{t+1}|x_t,d_t}$ * $\mathbf{V}$ eq. (41)
$\mathbf{T}_{x_{t+7},x_{t+3}|x_t,d_t}$ = $\mathbf{T}_{x_{t+6},x_{t+2}|x_t,d_t}$ * $\mathbf{V}$ eq. (41)
$\mathbf{T}_{x_{t+8},x_{t+4}|x_t,d_t}$ = $\mathbf{T}_{x_{t+7},x_{t+3}|x_t,d_t}$ * $\mathbf{V}$ eq. (41)
$\mathbf{T}_{x_{t+9},x_{t+5}|x_t,d_t}$ = $\mathbf{T}_{x_{t+8},x_{t+4}|x_t,d_t}$ * $\mathbf{V}$ eq. (41)
$\mathbf{T}_{x_{t+9},x_{t+5},x_t|x_t,d_t}$ = $\mathbf{T}_{x_{t+9},x_{t+5}|x_t,d_t}$ ⊙ $\mathbf{E}$ eq. (42)

Figure 10: Schematic representation of Algorithm 4. This example illustrates the HSMM with $n_x = 5$ and $n_d = 10$. The non-zero matrix elements are displayed as dots.

If we now apply step (42) it will give us $[\mathbf{A}_1^{(1)} \ \cdots \ \mathbf{A}_{n_d}^{(1)}] \in \mathbb{R}^{n_x^2 \times n_x n_d}$ with rank n_x^2 due to Lemma 7. Continuing in this manner, we can again repeatedly apply the step (41) to create a matrix with a rank at most n_x^2, since there are n_x^2 rows and assuming that $n_x n_d \geq n_x^2$. The number of times we need to apply (41) is now $n_x^2 - n_x$ since we need to go from n_x to n_x^2 independent columns.

In general, the step (41) needs to be applied $n_x^c - n_x^{c-1}$, in order to obtain n_x^c independent columns. The application of step (42) then creates $\mathbf{T}$ with rank n_x^{c+1}. Note, that since $\mathbf{T}$ has $n_x n_d$ columns, the maximum achievable rank is $n_x n_d$. ■

Observe that the above proof also provided the method for selecting the non-sequential observations $\mathbf{X}_{R_{t+1}}$. Specifically, since the set of observations $\mathbf{X}_{R_{t+1}} = \{o_{t+2}, \ldots\}$ must start from observation o_{t+2} and $|\mathbf{X}_{R_{t+1}}| = \ell$, we denote $s = t+2$. Then, ith added observation is $o_{s+(n_d-1)-(n_x^i-1)}$ for $i = 0, \ldots, \ell-2$ and the ℓth observation is $o_s = o_{t+2}$. For tensor $\underset{\mathbf{O}_{R_{t+1}|x_t d_t}}{\mathcal{F}}$ to achieve rank $n_x n_d$ we need to add $\ell = \lceil 1 + \frac{\log n_d}{\log n_x} \rceil$ observations.

Appendix B. Initial and Final Parts of HSMM

In this Section we present the derivations for the initial and final steps of HSMM, which were omitted from the main text. Specifically, this amounts to computing the factor $\mathcal{X}$ for two parts of the model, corresponding to $\mathbb{X}_{root}$ and $\mathbb{X}_T$ in Figures 11 and 12. The derivations for all other parts of HSMM were presented in the main text and this supplement.

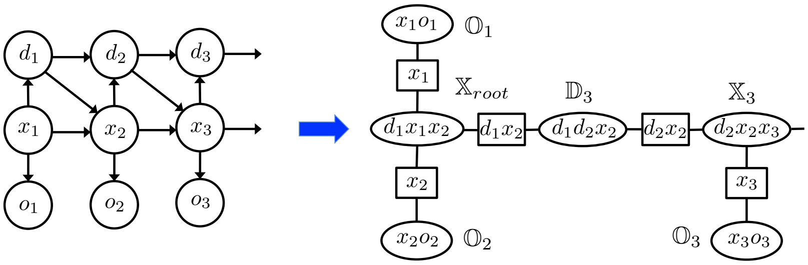

Figure 11: Part of HSMM corresponding to the initial time stamps and the related part of junction tree.

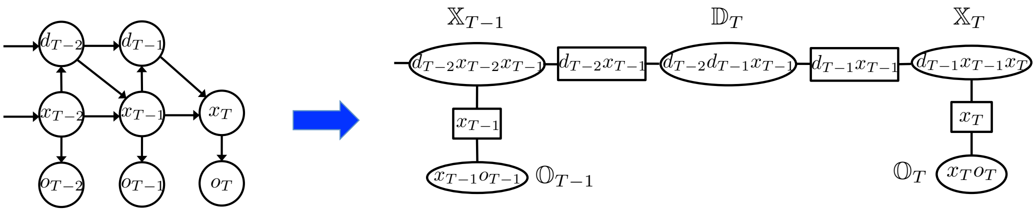

Figure 12: Part of HSMM corresponding to the final time stamps and the related part of junction tree.

To begin, recall the expression for the joint likelihood of the observed sequence:

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

and rewrite the above expression by keeping only the initial and final factors:

$$\begin{aligned}\underset{o_1,\ldots,o_T}{\mathcal{P}} &= \left(\underset{o_1|x_1}{\mathcal{O}} \times_{x_1} \left(\underset{x_2x_2|x_1d_1}{\mathcal{X}} \times_{x_2} \underset{o_2|x_2}{\mathcal{O}} \right) \right) \times_{x_2d_1} \underset{d_2|x_2x_2d_1}{\mathcal{D}} \times \cdots \\ &\cdots \times \underset{d_{T-1}|x_{T-1}x_{T-1}d_{T-2}}{\mathcal{D}} \times_{x_{T-1}d_{T-1}} \left(\underset{x_T|x_{T-1}d_{T-1}}{\mathcal{X}} \times_{x_T} \underset{o_T|x_T}{\mathcal{O}} \right). \qquad (45)\end{aligned}$$

Introduce the identity tensors into (45), regroup the terms and extract the factors $\mathcal{X}$:

$$\underset{\omega_{x_1}\omega_{x_2}\omega_{x_2d_1}}{\tilde{\mathcal{X}}} = \underset{\omega_{x_1}|x_1}{\mathcal{F}} \times_{x_1} \left(\underset{x_2x_2|x_1d_1}{\mathcal{X}} \times_{x_2} \underset{\omega_{x_2}|x_2}{\mathcal{F}} \right) \times_{x_2d_1} \underset{\omega_{x_2d_1}|x_2d_1}{\mathcal{F}} \qquad (46)$$

$$\underset{\omega_{x_{T-1}d_{T-1}}\omega_{x_T}}{\tilde{\mathcal{X}}} = \underset{\omega_{x_{T-1}d_{T-1}}|x_{T-1}d_{T-1}}{\mathcal{F}^{-1}} \times_{x_{T-1}d_{T-1}} \left(\underset{x_T|x_{T-1}d_{T-1}}{\mathcal{X}} \times_{x_T} \underset{\omega_{x_T}|x_T}{\mathcal{F}} \right). \qquad (47)$$

Defining the observable sets $\omega_{x_1} = o_1$, $\omega_{x_2} = o_2$ and $\omega_{x_2d_1} = \mathbf{o}_{R_3}$ we can rewrite (46) as follows:

$$\underset{o_1o_2\mathbf{o}_{R_3}}{\tilde{\mathcal{X}}} = \underset{o_1|x_1}{\mathcal{F}} \times_{x_1} \left(\underset{x_2x_2|x_1d_1}{\mathcal{X}} \times_{x_2} \underset{o_2|x_2}{\mathcal{F}} \right) \times_{x_2d_1} \underset{\mathbf{o}_{R_3}|x_2d_1}{\mathcal{F}}. \qquad (48)$$

Note that since all the factors participating in (48) are valid probability distributions, the resulting factor, i.e., $\underset{o_1o_2\mathbf{o}_{R_3}}{\tilde{\mathcal{X}}}$ is also a valid probability distribution, so it can be estimated directly from data. This is in contrast to the derivations we made for other parts of the model, where we had to perform additional transformations such as, for example in (10), in order to bring to the form, which could be estimated from the data samples.

In order to estimate (47), we compare it to the similar factor we considered in the main paper:

$$\underset{\omega_{x_{t-1}d_{t-1}}\omega_{x_t}\omega_{x_td_{t-1}}}{\tilde{\mathcal{X}}} = \underset{\omega_{x_{t-1}d_{t-1}}|x_{t-1}d_{t-1}}{\mathcal{F}^{-1}} \times_{x_{t-1}d_{t-1}} \left(\underset{x_tx_t|x_{t-1}x_{t-1}d_{t-1}}{\mathcal{X}} \times_{x_t} \underset{\omega_{x_t}|x_t}{\mathcal{F}} \right) \times_{x_td_{t-1}} \underset{\omega_{x_td_{t-1}}|x_td_{t-1}}{\mathcal{F}}, \qquad (49)$$

and observe that the last factor $\underset{\omega_{x_td_{t-1}}|x_td_{t-1}}{\mathcal{F}}$ in (49) is a conditional probability distribution, which has the following marginalization property

$$\underset{\omega_{x_td_{t-1}}|x_td_{t-1}}{\mathcal{F}} \times_{\omega_{x_td_{t-1}}} \underset{\omega_{x_td_{t-1}}}{\mathbf{1}} = \underset{x_td_{t-1}}{\mathbf{1}}, \qquad (50)$$

where $\mathbf{1}$ is the tensor, which has all elements equal to 1. The above can also be written in the scalar notations, $\sum_{\omega_{x_td_{t-1}}} p(\omega_{x_td_{t-1}}|x_td_{t-1}) = 1$ for each value of x_td_{t-1}. Therefore, if we apply (50) to (49), we get $\underset{\omega_{x_{t-1}d_{t-1}}\omega_{x_t}}{\tilde{\mathcal{X}}}$, which is the time-shifted version of $\underset{\omega_{x_{T-1}d_{T-1}}\omega_{x_T}}{\tilde{\mathcal{X}}}$. Therefore, to compute (47), we estimate the tensor in (13), i.e.,

$$\underset{\mathbf{o}_{R_t}o_t\mathbf{o}_{R_t}}{\tilde{\mathcal{X}}} = \underset{\mathbf{o}_{L_t}\mathbf{o}_{R_t}}{\mathcal{M}^{-1}} \times_{\mathbf{o}_{L_t}} \underset{\mathbf{o}_{L_t}\mathbf{o}_{R_t}o_t}{\mathcal{M}},$$

and marginalize out the right set of modes, corresponding to $\mathbf{O}_{R_t}$. Alternatively, we can use the batch estimate

$$\tilde{\mathfrak{X}} = \left(\sum_t \mathfrak{M}_{\mathbf{O}_{L_t}\mathbf{O}_{R_t}}\right)^{-1} \times_{\mathbf{O}_L} \left(\sum_t \mathfrak{M}_{\mathbf{O}_{L_t}\mathbf{O}_{R_t} o_t}\right),$$

and similarly perform the marginalization. This concludes our derivations.

References

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*, 2013a.

A. Anandkumar, D. Hsu, M. Janzamin, and S. M. Kakade. When are overcomplete topic models identifiable? Uniqueness of tensor Tucker decompositions with structured sparsity. In *Advances in Neural Information Processing Systems*, pages 1986–1994, 2013b.

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, 2013c.

A. Anandkumar, R. Ge, D. Hsu, and S. M. Kakade. A tensor approach to learning mixed membership community models. *Journal of Machine Learning Research*, 15:2239–2312, 2014a.

A. Anandkumar, R. Ge, D. Hsu, S. M. Kakade, and M. Telgarsky. Tensor decompositions for learning latent variable models. *Journal of Machine Learning Research*, 15(1):2773–2832, 2014b.

R. Bailly, F. Denis, and L. Ralaivola. Grammatical inference as a principal component analysis problem. In *Proceedings of the International Conference on Machine Learning*, pages 33–40, 2009.

B. Balle, A. Quattoni, and X. Carreras. A spectral learning algorithm for finite state transducers. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 156–171, 2011.

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

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.

B. Boots and G. J. Gordon. Predictive state temporal difference learning. In *Advances in Neural Information Processing Systems*, pages 271–279. 2010.

S. Budalakoti, A. N. Srivastava, and M. E. Otey. Anomaly detection and diagnosis algorithms for discrete symbol sequences with applications to airline safety. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 39(1):101–113, 2009.

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

S. B. Cohen, K. Stratos, M. Collins, D. P. Foster, and L. Ungar. Spectral learning of latent-variable PCFGs: Algorithms and sample complexity. *Journal of Machine Learning Research*, 15:2399–2449, 2014.

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.

P. Dhillon, D. P. Foster, and L. H. Ungar. Multi-view learning of word embeddings via CCA. In *Advances in Neural Information Processing Systems*, pages 199–207, 2011.

E. B. Fox, E. B. Sudderth, M. I. Jordan, and A. S. Willsky. An HDP-HMM for systems with state persistence. In *Proceedings of the International Conference on Machine Learning*, pages 312–319, 2008.

D. Gorinevsky, B. Matthews, and R. Martin. Aircraft anomaly detection using performance models trained on fleet data. In *Proceedings of the Conference on Intelligent Data Understanding*, pages 17–23, 2012.

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.

H. Jaeger. Observable operator models for discrete stochastic time series. *Neural Computation*, 12 (6):1371–1398, 2000.

M. J. Johnson and A. S. Willsky. Bayesian nonparametric hidden semi-Markov models. *Journal of Machine Learning Research*, 14:673–701, 2013.

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

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

D. A. Levin, Y. Peres, and E. Wilmer. *Markov chains and mixing times*. American Mathematical Society, 2009.

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.

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.

I. Melnyk, B. Matthews, H. Valizadegan, A. Banerjee, and N. Oza. Vector autoregressive model-based anomaly detection in aviation systems. *Journal of Aerospace Information Systems*, pages 161–173, 2016.

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.

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

NASA. Flight data set. Available at https://c3.nasa.gov/dashlink/projects/85/.

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.

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

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

S. M. Siddiqi, B. Boots, and G. J. Gordon. Reduced-rank hidden Markov models. In *Proceedings of the Artificial Intelligence and Statistics Conference*, pages 741–748, 2010.

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

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.

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.

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

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.

H. Zen, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura. A hidden semi-Markov model-based speech synthesis system. *Transactions on Information Systems*, E90-D(5):825–834, 2007.