

MIT Open Access Articles

Minimal Realization Problems for Hidden Markov Models

The MIT Faculty has made this article openly available. **Please share** how this access benefits you. Your story matters.

Citation: Huang, Qingqing, Rong Ge, Sham Kakade, and Munther Dahleh. "Minimal Realization Problems for Hidden Markov Models." IEEE Transactions on Signal Processing 64, no. 7 (April 2016): 1896–1904.

As Published: <http://dx.doi.org/10.1109/tsp.2015.2510969>

Publisher: Institute of Electrical and Electronics Engineers (IEEE)

Persistent URL: <http://hdl.handle.net/1721.1/110794>

Version: Original manuscript: author's manuscript prior to formal peer review

Terms of use: Creative Commons Attribution-Noncommercial-Share Alike



Minimal Realization Problems for Hidden Markov Models

Qingqing Huang[†], Rong Ge[‡], Sham Kakade[‡], Munther Dahleh[†]

Abstract

This paper addresses two fundamental problems in the context of Hidden Markov Models (HMMs). The first problem is concerned with the characterization and computation of a minimal order HMM that realizes the exact joint densities of an output process based on only finite strings of such densities (known as HMM partial realization problem). The second problem is concerned with learning a HMM from finite output observations of a stochastic process. We review and connect two fields of studies: realization theory of HMMs, and the recent development in spectral methods for learning latent variable models. Our main results in this paper focus on generic situations, namely, statements that will be true for almost all HMMs, excluding a measure zero set in the parameter space. In the main theorem, we show that both the minimal quasi-HMM realization and the minimal HMM realization can be efficiently computed based on the joint probabilities of length N strings, for N in the order of $\mathcal{O}(\log_d(k))$. In other words, learning a quasi-HMM and an HMM have comparable complexity for almost all HMMs.

I. INTRODUCTION

A. Background

Hidden Markov Models (HMMs) are widely used for describing discrete random processes, especially in the applications involving temporal pattern recognition such as speech and gesture recognition, part-of-speech tagging and parsing, and bioinformatics. The Markovian property of the hidden state evolution potentially leads to a low complexity representation of the output random process. In this work, we consider the long-standing HMM realization problem: given some partial knowledge about the output process of an unknown HMM, can we generalize it to a full description of the random process?

Consider a discrete random process $\{y_t : t \in \mathbb{Z}\}$, which assumes values in a finite alphabet $[d] \equiv \{1, \dots, d\}$. Assume that y_t is the output process of a stationary HMM of finite order. Let the random vector $\mathbf{y}_1^N = (y_1, \dots, y_N)$ denote an string of length N , which assumes values in the N -ary Cartesian product $[d]^N$. The process y_t is fully characterized by the joint probabilities of strings of any length in the countably infinite table (denoted by $\mathcal{P}^{(\infty)}$):

$$\left\{ \mathbb{P}(y_1 = l_1, \dots, y_N = l_N) : \forall \mathbf{l}_1^N \in [d]^N, \forall N \in \mathbb{Z} \right\}.$$

There are three main concerns in the realization problem:

- 1) **(Informational complexity)** Suppose that the underlying HMM is of order k , and we are given the joint probabilities of all the length N strings, namely:

$$\mathcal{P}^{(N)} \equiv \left\{ \mathbb{P}(y_1 = l_1, \dots, y_N = l_N) : \forall \mathbf{l}_1^N \in [d]^N \right\},$$

how large does N need to be so that we can compute $\mathcal{P}^{(\infty)}$ based on $\mathcal{P}^{(N)}$?

- 2) **(Computational complexity)** Can we solve the realization problem with runtime polynomial in the dimensions (alphabet size d and order of the underlying HMM k)?
- 3) **(Statistical complexity)** When $\mathcal{P}^{(N)}$ is estimated from sample sequences and has some estimation error, are the realization algorithms robust to the input errors?

These are long standing questions, and there are several lines of work within different communities at tempting to address these questions. It has long been known that, in the information theoretic sense, there exist hard cases of HMMs that are not efficiently PAC learnable [13] [17]. However, a more practical question is, can we efficiently solve the realization / learning problem for most HMMs? In this work, we focus on generic analysis and show that,

Copyright (c) 2015 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

Q. Huang, and M. Dahleh are with the Laboratory of Information and Decision Systems, Massachusetts Institute of Technology (Emails: {qqh, dahleh}@mit.edu); R.Ge and S.Kakade are with Microsoft Research, New England (Emails: {rongge, skakade}@microsoft.com).

for almost all HMMs, i.e., excluding those whose parameters are in a measure zero set¹, the realization problems can be efficiently solved with poly time algorithms.

B. Organization

To study the HMM realization problems, we focus on algorithms based on the spectral method. The basic idea is to exploit the recursive structural properties of the underlying finite state model, and write the joint probabilities in $\mathcal{P}^{(N)}$ into a specific form which admits *rank decomposition*, where the rank reveals the minimal order of the realization and the model parameters can be extracted from the factors.

In the first part (Section III), we consider the problem of finding the minimal quasi-HMM realization. Quasi-HMMs are associated with different names in different communities, for example finite state regular automata [4], [5], regular quasi realization [17], [21], and operator models [10], [17]. We mostly follow the terminologies in [21]. Algorithm 1 is the well-known algorithm for finding the minimal order quasi-HMM realization (to be rigorously defined later). However, in general the window size N can not be specified a priori and thus the complexity of the algorithm cannot be explicitly determined. In Theorem 1, we show that, if the output process is generated by an general position HMM with order k , we only need the window size N in the order of $\mathcal{O}(\log_d(k))$ for pinning down $\mathcal{P}^{(\infty)}$ based on $\mathcal{P}^{(N)}$, where d is the output alphabet size. Moreover, we show that Algorithm 1 has runtime and sample complexity both polynomial in the relevant parameters.

In the second part (Section IV), we consider the problem of finding the minimal HMM realization, using tensor decomposition methods, which rely on the uniqueness of tensor decomposition to recover the minimal order HMM that is unique up to hidden states permutation. Tensor decomposition based algorithms for learning HMMs are studied in [1], [2], [6]. In these works, the transition matrix is always assumed to be of full rank. Similar to that in the quasi-HMM realization problem, in general the required window size N and also the complexity of the algorithm cannot be determined a priori. In [1], the authors examined the generic identifiability conditions of HMM, and showed that generically it suffices to pick the window size $N = 2n + 1$ for some positive integer n , such that $\binom{n+d-1}{d-1} \geq k$. In the case where d is much smaller than k , n needs to be in the order of $\mathcal{O}(k^{1/d})$. Another bound on the window size N is given in [6], which is in the order of $\mathcal{O}(k/d)$. However, the size of the tensor in the decomposition is exponential in n , thus all these bound lead to runtime exponential in k .

In Section IV, we propose a two-step realization approach, and analyze the identifiability issue of the two steps. Then, we show that for the processes generated by almost all HMMs, the window size N only needs to be in the order of $\mathcal{O}(\log_d(k))$ for finding the minimal HMM realization. This means that for most HMMs, finding minimal quasi-HMM and minimal HMM realizations are actually of equal difficulty.

II. MINIMAL REALIZATION PROBLEM FORMULATION

In this section, we first review the basics of HMMs, and then formally introduce the quasi-HMM and HMM realization problems.

A. Preliminaries on HMMs

An HMM determines the joint probability distribution over sequences of hidden states $\{x_t : t \in \mathbb{Z}\}$ and observations $\{y_t : t \in \mathbb{Z}\}$. For simplicity, we call each output y_t as a “letter” taking value from some discrete alphabet $[d]$, and a sequence of n letters is referred to as a “string”, taking value from the Cartesian product $[d]^n$. We use $[d^N] \equiv \{1, \dots, d^N\}$ to denote the vectorized indices in $[d]^n$.

The joint distribution of $\{x_t, y_t : t \in \mathbb{Z}\}$ from a stationary HMM is parameterized by a pair of matrices: the state transition matrix $Q \in \mathbb{R}_+^{k \times k}$, and the observation matrix $O \in \mathbb{R}_+^{d \times k}$, which satisfy $\mathbf{e}^\top O = \mathbf{e}^\top$ and $\mathbf{e}^\top Q = \mathbf{e}^\top$, where \mathbf{e} is the all ones vector. The hidden state x_t evolves following a Markov process:

$$\mathbb{P}(x_{t+1} = j | x_t = i) = Q_{j,i}.$$

Let π denote the stationary state distribution, i.e., $\pi_i = \mathbb{P}[x_t = i]$ and $Q\pi = \pi$. Without loss of generality, we assume that $\pi_i > 0$ for all $i \in [k]$. We also define the backward transition matrix $\tilde{Q} \in \mathbb{R}^{k \times k}$:

$$\mathbb{P}(x_{t-1} = j | x_t = i) = \tilde{Q}_{j,i}.$$

¹ In our setting, algebraic genericity coincides with the measure theoretic notion of generic. Throughout the discussion, for fixed alphabet size d and order k , we call an HMM in general position if its transition and observation matrix are in general position, which is equivalent to “almost everywhere in the parameter space of $\{Q \in \mathbb{R}_+^{k \times k}, O \in \mathbb{R}_+^{d \times k} : \mathbf{e}^\top Q = \mathbf{e}^\top, \mathbf{e}^\top O = \mathbf{e}^\top\}$ ”.

Observe that the matrix \tilde{Q} is related to Q as: $\tilde{Q} = \text{Diag}(\pi)Q^\top \text{Diag}(\pi)^{-1}$. Conditioned on the hidden state taking value i , the probability of observing letter j is:

$$\mathbb{P}(y_t = j | x_t = i) = O_{j,i}.$$

We call two HMMs equivalent if the output processes are statistically indistinguishable.

The order of the HMM is defined to be the number of hidden states, denoted by k . We will denote the class of all HMMs with output alphabet size d and order k by $\Theta_{(d,k)}^h$.

B. Problem formulations

The *realization problem* takes as inputs the probabilities of finite length strings for a fixed window size N ($\mathcal{P}^{(N)}$), and finds a finite state model of the minimal order to describe the entire output process ($\mathcal{P}^{(\infty)}$). We aim to find the most succinct description of the process, namely the minimal order realization, where the “order” refers to the number of states of the underlying finite state model. Without loss of generality, we assume that the process has a minimal realization of order k and examine under what conditions the algorithms can recover an equivalent minimal order realization.

Next, we introduce two classes of finite state models, both of which can realize an HMM output process.

Definition 1 (Quasi-HMM realization [21]). *Let θ^o be a tuple: $\theta^o = (k, u, v \in \mathbb{R}^k, A^{(j)} \in \mathbb{R}^{k \times k} : \forall j \in [d])$. We call θ^o a quasi-HMM realization of order k for a stationary process $\{y_t : t \in \mathbb{Z}\}$ if the three conditions hold: ($\forall \mathbf{1}_1^N \in [d]^N, \forall N \in \mathbb{Z}$)*

$$\mathbb{P}(\mathbf{y}_1^N = \mathbf{1}_1^N) = u^\top A^{(l_1)} A^{(l_2)} \dots A^{(l_N)} v, \quad (1)$$

$$u^\top \left(\sum_{j=1}^d A^{(j)} \right) = u^\top, \quad (2)$$

$$\left(\sum_{j=1}^d A^{(j)} \right) v = v. \quad (3)$$

Definition 2 (Equivalent quasi-HMM realizations). *Two quasi-HMM realizations $\theta^o = (k, u, v, A^{(j)} : j \in [d])$ and $\tilde{\theta}^o = (k, \tilde{u}, \tilde{v}, \tilde{A}^{(j)} : j \in [d])$ are called equivalent, if there is a full rank matrix $T \in \mathbb{R}^{k \times k}$ such that:*

$$\tilde{u} = T^\top u, \quad \tilde{v} = T^{-1} v, \quad \tilde{A}^{(j)} = T^{-1} A^{(j)} T, \quad \forall j \in [d].$$

Definition 3 (HMM realization). *Let θ^h be a tuple: $\theta^h = (k, O \in \mathbb{R}_+^{d \times k}, Q \in \mathbb{R}_+^{k \times k})$. We call θ^h an HMM realization of order k for a stationary random process $\{y_t : t \in \mathbb{Z}\}$, if the matrices Q and O are column stochastic, and the output process of the HMM defined by the transition matrix Q and observation matrix O has the same distribution as y_t .*

HMM realizations are in a subset of the model class of quasi-HMM realizations. Given an HMM realization $\theta^h = (k, O, Q)$, one can construct the following quasi-HMM realization $\theta^o = (k, u, v, A^{(j)} : j \in [d])$:

$$u = \mathbf{e}, \quad (4)$$

$$v = \pi, \quad (5)$$

$$A^{(j)} = Q \text{Diag}(O_{[j,:]}), \quad \forall j \in [d]. \quad (6)$$

The minimal (quasi-)HMM realization problem is formally stated below: Assume that the random process is the output of an HMM of order k . How large does the window size N need to be, so that given the joint probabilities $\mathcal{P}^{(N)}$ we can efficiently construct a minimal (quasi-)HMM realization for the process?

III. MINIMAL QUASI-HMM REALIZATION

In this section, we address the minimal quasi-HMM realization problem. We first review the widely used algorithm [3], [5]; then we show for HMMs in general position, the window size N only needs to be in the order of $\mathcal{O}(\log_d(k))$ to guarantee the correctness of the algorithm; we also give an example of hard case (degenerate) which needs N to be as large as k ; finally we examine the stability of the algorithm.

A. Algorithm

For notational convenience, we define the bijective mapping $L : [d]^n \rightarrow [d]^n$ which maps the multi-index $\mathbf{l}_1^n = (l_1, \dots, l_n) \in [d]^n$ to the index $L(\mathbf{l}_1^n) = (l_1 - 1)d^{n-1} + (l_2 - 1)d^{n-2} + \dots + l_n \in [d^n]$.

Given the length N joint probabilities $\mathcal{P}^{(N)}$, where $N = 2n + 1$ for some positive number n , we form two matrices $H^{(0)}, H^{(j)} \in \mathbb{R}^{d^n \times d^n}$ for all $j \in [d]$ as below:

$$[H^{(0)}]_{L(\mathbf{l}_1^n), L(\mathbf{l}_{-1}^n)} = \mathbb{P}(\mathbf{y}_{-1}^{-n} = \mathbf{l}_{-1}^{-n}, \mathbf{y}_0^{n-1} = \mathbf{l}_1^n), \quad (7)$$

$$[H^{(j)}]_{L(\mathbf{l}_1^n), L(\mathbf{l}_{-1}^n)} = \mathbb{P}(\mathbf{y}_{-1}^{-n} = \mathbf{l}_{-1}^{-n}, y_0 = j, \mathbf{y}_1^n = \mathbf{l}_1^n), \quad (8)$$

where $\mathbf{l}_1^n = (l_1, \dots, l_n)$ and $\mathbf{l}_{-1}^n = (l_{-1}, l_{-2}, \dots, l_{-n}) \in [d]^n$ denotes the length n string corresponding to the future and the past n time slots, respectively. Note that the “future” observations and the “past” observations are independent conditioned on the “current” state, which is the Markovian property that Algorithm 1 relies on.

Algorithm 1 Minimal quasi-HMM realization

Input: $H^{(0)}, H^{(j)} \in \mathbb{R}^{d^n \times d^n}$ for all $j \in [d]$

Output: $\tilde{\theta}^o = (k, \tilde{u}, \tilde{v}, \tilde{A}^{(j)} : j \in [d])$

1) Compute the SVD of $H^{(0)}$:

$$H^{(0)} = U_H D_H V_H'. \quad (9)$$

Set $U = U_H D_H^{1/2}$, $V = V_H D_H^{1/2}$.

2) Let \tilde{k} be the rank of $H^{(0)}$, and let

$$\tilde{u} = U' \mathbf{e}, \quad \tilde{v} = V' \mathbf{e}. \quad (10)$$

3) Let U^\dagger and V^\dagger be the pseudo inverse of U and V .

$$\tilde{A}^{(j)} = U^\dagger H^{(j)} (V^\dagger)', \quad \forall j \in [d]. \quad (11)$$

The core idea of Algorithm 1 was discussed in [11], and it has been rediscovered numerous times in the literature in slightly different forms [3], [5]. We summarize the main idea below.

Remark 1 (Minimal order). Let $\theta^o = (k, u, v, A^{(j)} : j \in [d])$ be a minimal quasi-HMM realization of order k for the process considered. Since the joint probabilities can be factorized in terms of the $A^{(j)}$'s as in (1), one can factorize $H^{(0)}$ and $H^{(j)}$'s as below:

$$H^{(0)} = E F^\top, \quad H^{(j)} = E A^{(j)} F^\top,$$

where the matrices $E, F \in \mathbb{R}^{d^n \times k}$ are functions of θ^o . In particular, the $L(\mathbf{l}_1^n)$ -th row of E and F are given by:

$$E_{[L(\mathbf{l}_1^n), :]} = u^\top (A^{(l_n)} \dots A^{(l_1)}), \quad (12)$$

$$F_{[L(\mathbf{l}_1^n), :]} = v^\top (A^{(l_n)} \dots A^{(l_1)})^\top. \quad (13)$$

Note that if both E and F have full column rank k , then $H^{(0)}$ has rank k , according to Sylvester's inequality. Any rank factorization leads to an equivalent minimal quasi-HMM realization of order k . The minimal order condition, though not explicitly enforced, is reflected in the rank factorization, as any quasi-HMM realization of lower order results in a matrix $H^{(0)}$ of lower rank, which leads to a contradiction.

The correctness of the algorithm crucially relies on matrix $H^{(0)}$ achieving its maximal rank k , which equals the order of the minimal realization. A necessary condition for the correctness of the algorithm is stated below.

Lemma 1 (Correctness of Algorithm 1). Assume the process has a minimal quasi-realization θ^o of order k . Algorithm 1 returns a minimal quasi-HMM realization $\tilde{\theta}^o$ that is equivalent to θ^o , if the matrices E, F defined in (12) and (13) have full column rank k .

Increasing the window size N can potentially boost the rank of $H^{(0)}$, in the hope that the $H^{(0)}$ reaches its maximal rank and Algorithm 1 can correctly find the minimal realization. However, for a given random process,

the study of [19] showed that it is undecidable to verify whether it has a *finite order* quasi-HMM realization. Even under our assumption that the process indeed has an order k minimal quasi-HMM realization, it is still not clear how large the size of matrix $H^{(0)}$ ($d^n \times d^n$) needs to be so that it achieves the maximal rank k . In previous works, it was usually implicitly assumed that N is large enough so that $H^{(0)}$ achieves its maximal rank [5]. Yet without a bound on n or N the computational complexity of the algorithm is ambiguous.

B. Generic analysis of information complexity

We desire a small window size N while guaranteeing the full column rank of the matrices E and F defined in (12) and (13). The following theorem shows that if the random process is generated by an order k HMM in general position, then we only need window size $N > 4\lceil \log_d(k) \rceil + 1$ to guarantee the correctness of Algorithm 1.

Theorem 1 (Window size N for quasi-HMM).

- (1) Consider $\Theta_{(d,k)}^h$, the class of all HMMs with output alphabet size d and order k . There exists a measure zero set $\mathcal{E} \in \Theta_{(d,k)}^h$, such that for all the output process generated by HMMs in $\Theta_{(d,k)}^h \setminus \mathcal{E}$, Algorithm 1 returns a minimal quasi-HMM realization, if window size $N = 2n + 1$ for some n such that:

$$n > 8\lceil \log_d(k) \rceil. \quad (14)$$

- (2) For any pair of (d, k) , randomly pick an instance from the class $\Theta_{(d,k)}^h$. If for a given window size $N = 2n + 1$, the matrix $H^{(0)}$ achieves its maximal rank k , then for all HMMs in $\Theta_{(d,k)}^h$, excluding a measure zero set, N is sufficiently large for the correctness of Algorithm 1.

Since the elements of matrices E and F are polynomials of the parameters Q and O , in order to show E has full column rank for Q and O in general position, it suffices to construct an instance of HMM for which the matrix E has full column rank. In particular, we fix the transition matrix Q and randomize the observation matrix O and bound the singular values of E in probability. The detailed proof is provided in Appendix B.

For all (d, k) pairs in the set $\{2 \leq d \leq k < 3000\}$, we implemented the test in Theorem 1 (2), and found that for all these cases $n = \lceil \log_d(k) \rceil$ is sufficient. We conjecture that in general, $n \geq \log_d(k)$ is enough.

In the worst case [21], the ‘‘Hankel rank’’ of the matrix $H^{(0)}$ with infinite window size can be larger than the rank of any finite size block of the infinite matrix. Instead of the worst case analysis, our generic analysis examines the average cases, and it has the following implications: if the process is generated by some average case HMM of order k , then the Hankel rank equals k ; moreover, the window size n only needs to be in the order of $O(\log_d(k))$ so that the rank of finite matrix $H^{(0)}$ achieves the Hankel rank.

C. Existence of hard cases

We showed that for generic HMM output processes, Algorithm 1 has polynomial runtime. There exists a long line of hardness results for learning HMMs [13], [17], [20], showing that in the worst case (lie in the measure zero set in the parameter space) learning the distribution of an HMM *can* be computationally hard under cryptographic assumptions.

In Fig. 1, we adapt the hardness results to our setting and give an example. The state diagram describes the transition and observation probabilities. Solving the realization problem is equivalent to learning the joint distribution of the process. One can verify that the window size N needs to be at least as large as T , which is proportional to the order of the underlying HMM, and therefore the computation complexity is exponential in the order of the HMM.

We point out that not all HMMs in the measure zero set are information theoretically hard to learn. For instance, consider the degenerate HMM in [1] with the transition matrix $Q = I_{k \times k}$ and with general position observation matrix O . Suppose that $d \ll k$, it was shown that the window size N needs to be in the order of $k^{\frac{1}{d}}$ so that matrices E and F attain full column rank. However the distribution of this i.i.d. process is not fundamentally difficult to learn. It remains an open problem to find realization algorithm that can handle more cases.

D. Stability analysis

In practice, the joint probabilities in $\mathcal{P}^{(N)}$ are estimated based on finite sample sequences of the process. In the next theorem, we show that in order to achieve ϵ -accuracy in the parameters of the minimal quasi-HMM realization,

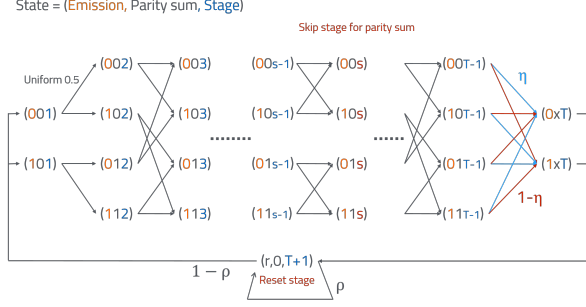


Fig. 1. Reduction of HMM to noisy parity to lower bound the worst case computational complexity. In the state transition diagram, for stage $t = 1, \dots, T-1$, the emission state E_t is uniformly distributed over $\{0, 1\}$ and is observed. For stage $t = 2, \dots, T-1$, the parity state S_t computes $E_{t-1} \oplus S_{t-1}$, except for at one unknown stage s , $S_t = S_{t-1}$. At stage T , with probability η , the correct parity state S_{T-1} is revealed, and with probability $1 - \eta$, the complement is observed. $(T+1)$ is a reset stage, with probability ρ it stays in the reset stage.

the number of sample sequences we need to estimate $\mathcal{P}^{(N)}$ is polynomial in all relevant parameters, including the order k .

Theorem 2. Given T independent sample sequences of the output process of an HMM of order k and with alphabet size d . Construct $\hat{H}^{(0)}$ and $\hat{H}^{(j)}$'s as in (7) and (8) with the empirical probabilities. Let $N = 2n + 1$, and $n = 2\lceil \log_d(k) \rceil$. Let $\hat{\theta}^o = (k, \tilde{u}, \tilde{v}, \hat{A}^{(j)} : j \in [d])$ and $\tilde{\theta}^o = (k, \tilde{u}, \tilde{v}, \tilde{A}^{(j)} : j \in [d])$ be the output of Algorithm 1 with the empirical probabilities and the exact probabilities for the input, respectively. Then, in order to achieve ϵ -accuracy in the output with probability at least $1 - \eta$, namely:

$$\|\hat{u} - \tilde{u}\| \leq \epsilon, \|\hat{v} - \tilde{v}\| \leq \epsilon, \|\hat{A}^{(j)} - \tilde{A}^{(j)}\| \leq \epsilon, \forall j,$$

the number of independent sample sequences we need is given by:

$$T = \frac{Ck^6 d^4}{\epsilon^4 \sigma_k^8} \log \left(\frac{2k^4 d^2}{\eta} \right),$$

where σ_k is the k -th singular value of $H^{(0)}$ and C is some absolute constant.

Since the core of the algorithm is singular value decomposition of the matrix $H^{(0)}$, the stability analysis mostly uses the standard matrix perturbation results, which we review in Appendix C. The detailed proof is provided in Appendix B.

Remark 2. Note that Theorem 1 shows that for window size N large enough ($O(\log_d(k))$), the exact realization problem (no estimation noise) can be solved with poly time algorithm. When empirical probabilities are used, Theorem 2 shows that the required number of independent samples is polynomial in k , d , and $1/\sigma_k$. σ_k depends on the HMM that generates the process. In the proof of Theorem 1, it is showed that there exist cases for which σ_k is lower bounded by constant, for which case the sample complexity is indeed polynomial; however there also exists hard cases for which σ_k is arbitrarily small. We defer the analysis of sample complexity, which relies on understanding the relation between window size, HMM parameter, and σ_k , to future work.

IV. MINIMAL HMM REALIZATION PROBLEM

Recall that an HMM can be easily converted to a quasi-HMM of the same order as shown in (4)–(6), yet given a quasi-HMM realization it is difficult to construct an HMM [3]. In this section, we apply tensor decomposition techniques to study the minimal HMM realization problem and discuss its connection to the previous section. In particular, we show that for processes generated by general position HMMs, the two realization problems have similar computational complexity.

A. Preliminaries on tensor algebra

a) *Definitions:* Tensor algebra has many similarities to but also many striking differences from matrix algebra, one of which is that, under very mild conditions, tensor *minimal rank* decomposition is unique up to column scaling

and permutation, which is the key property exploited to uniquely identify the *minimal* HMM realization. This is in parallel with the fact that we use matrix rank decomposition to find a minimal quasi-HMM realization.

We review some properties of 3rd order tensors below. A more detailed introduction to tensor algebra can be found in [14] and the references therein. One way to view a 3rd order tensor $X \in \mathbb{R}^{n_A \times n_B \times n_C}$ is that it defines a three-way array, multi-indexed by (j_1, j_2, j_3) , $\forall j_1 \in [n_A], j_2 \in [n_B], j_3 \in [n_C]$. A rank-1 tensor $X = a \otimes b \otimes c$ is defined to be the outer-product of the three vectors a, b, c and $X_{j_1, j_2, j_3} = a_{j_1} b_{j_2} c_{j_3}$. Tensor rank decomposition is a natural extension of matrix singular value decomposition (SVD) to higher order tensors.

Definition 4 (Tensor rank decomposition). *The rank decomposition of a 3rd order tensor $X \in \mathbb{R}^{n_A \times n_B \times n_C}$ is a sum of rank-1 tensors for the smallest number of summands k :*

$$X = A \otimes B \otimes C = \sum_{i=1}^k A_{[:,i]} \otimes B_{[:,i]} \otimes C_{[:,i]},$$

where matrices $A \in \mathbb{R}^{n_A \times k}$, $B \in \mathbb{R}^{n_B \times k}$, $C \in \mathbb{R}^{n_C \times k}$. The minimal number of summands k is defined to be the rank of the tensor.

A tensor can also be viewed as a multi-linear operator. Consider a 3rd order tensor X . For given m_A, m_B, m_C , it defines a multi-linear mapping $X(V_1, V_2, V_3) : \mathbb{R}^{m_A \times n_A} \times \mathbb{R}^{m_B \times n_B} \times \mathbb{R}^{m_C \times n_C} \rightarrow \mathbb{R}^{m_A \times m_B \times m_C}$ as below: ($\forall j_1 \in [m_A], j_2 \in [m_B], j_3 \in [m_C]$)

$$\begin{aligned} & [X(V_1, V_2, V_3)]_{j_1, j_2, j_3} \\ &= \sum_{i_1 \in [n_A], i_2 \in [n_B], i_3 \in [n_C]} X_{i_1, i_2, i_3} [V_1]_{j_1, i_1} [V_2]_{j_2, i_2} [V_3]_{j_3, i_3}. \end{aligned} \quad (15)$$

Assuming that the tensor admits a decomposition $X = A \otimes B \otimes C \in \mathbb{R}^{n_A \times n_B \times n_C}$, we can equivalently write:

$$X(V_1, V_2, V_3) = (V_1 A) \otimes (V_2 B) \otimes (V_3 C), \quad (16)$$

Note that X can have different forms of decompositions, yet the mappings defined in (16) are all equivalent.

Definition 5 (Khatri-Rao product). *For matrices $A \in \mathbb{R}^{n_A \times k}$, $B \in \mathbb{R}^{n_B \times k}$, the (column) Khatri-Rao product $X = A \odot B \in \mathbb{R}^{n_A n_B \times k}$ is defined as follows:*

$$X_{(j_1-1)n_B+j_2, i} = A_{j_1, i} B_{j_2, i},$$

and each column of X is a rank-1 Khatri-Rao product.

An equivalent representation of a 3rd order tensor $X \in \mathbb{R}^{n_A \times n_B \times n_C}$ is given by its matricization, obtained by rearranging the elements of the tensor into a matrix. For example, the matricization along the third mode gives a matrix $\overline{X}^{(3)}$ is specified as below:

$$\left[\overline{X}^{(3)} \right]_{j_3, ((j_1-1)n_B+j_2)} = X_{j_1, j_2, j_3}.$$

Moreover, if the tensor admits a decomposition $X = A \otimes B \otimes C$, we can write the matricization as Khatri-Rao product of the factors: $\overline{X}^{(3)} = C(A \odot B)^\top$.

b) Uniqueness condition: Unlike the rank decomposition of matrices, under rather mild conditions of the factors we can uniquely (up to common column permutation and scaling) identify the factors from the 3rd order tensor X . In the following, we state a set of sufficient conditions on the factors A, B, C that guarantee the uniqueness of tensor decomposition,

Definition 6 (Kruskal rank). *The Kruskal rank of a matrix $A \in \mathbb{R}^{n \times m}$ equals r if any set of r columns of A are linearly independent, and there exists a set of $(r+1)$ columns that are linearly dependent (if $r < m$).*

Lemma 2 (Uniqueness of tensor decomposition ([15], [18])). *The tensor factorization $X = A \otimes B \otimes C$ is unique up to column permutation and scaling, if*

$$\text{krank}(A) + \text{krank}(B) + \text{krank}(C) \geq 2k + 2. \quad (17)$$

c) *Decomposition algorithms*: Unlike matrix SVD, in general tensor decomposition is a hard problem [14]. Nevertheless, for cases where the factors A, B, C satisfy certain rank conditions, there exist efficient and provable algorithms. We include the detailed steps of the algorithm in the appendix for completeness.

If the matrix A and B both have full column rank, Algorithms 3 in the appendix ([16]) can uniquely recover the factors up to common column permutation, with running time polynomial in the dimension of the tensor. Other algorithms such as tensor power method and recursive projection, which are possibly more stable in practice, also apply here.

Algorithm 4 is another efficient tensor decomposition algorithm ([8] [12]) to a subset of the degenerate instances whose transition matrix is rank deficient. Instead of requiring both A and B to be of full rank k , this algorithm requires that the factor C and the Khatri-Rao product $A \odot B$ have full column rank k . The basic idea of the algorithm is as follows: there is a *unique rank decomposition* of the 3rd dimension matricization of the tensor: $\bar{X}^{(3)} = FE^\top = C(A \odot B)^\top$, under the algebraic constraints that each column of the matrix E is a rank one Khatri-Rao product.

B. Minimal HMM realization

d) *Formulation*: For a fixed window size $N = 2n + 1$, given the exact joint probabilities in $\mathcal{P}^{(N)}$, similar to the construction of $H^{(0)}$ in (7), one can construct a 3rd order tensor $M \in \mathbb{R}^{d^n \times d^n \times d}$ as below:

$$M_{L(\mathbf{1}_1^n), L(\mathbf{1}_{-1}^{-n}), l_0} = \mathbb{P}(\mathbf{y}_{-n}^n = \mathbf{1}_{-n}^n), \quad \forall \mathbf{1}_{-n}^n \in [d]^N. \quad (18)$$

Suppose that the process has a minimal HMM realization $\theta^h = (k, Q, O)$ of order k . We can write M as a tensor product:

$$M = A \otimes B \otimes C, \quad (19)$$

where the matrices $A, B \in \mathbb{R}^{d^n \times k}$ and $C \in \mathbb{R}^{d \times k}$ correspond to the conditional probabilities:

$$A_{L(\mathbf{1}_1^n), m} = \mathbb{P}(\mathbf{y}_1^n = \mathbf{1}_1^n | x_0 = m), \quad (20)$$

$$B_{L(\mathbf{1}_{-1}^{-n}), m} = \mathbb{P}(\mathbf{y}_{-1}^{-n} = \mathbf{1}_{-1}^{-n} | x_0 = m), \quad (21)$$

$$C_{l, m} = \mathbb{P}(y_0 = l, x_0 = m). \quad (22)$$

Moreover, observe that A and B are recursive linear functions of the model parameters Q and O as below:

$$A^{(n)} = \mathbb{P}(\mathbf{y}_1^n | x_0 = m) = (O \odot A^{(n-1)})Q, \quad (23)$$

$$B^{(n)} = \mathbb{P}(\mathbf{y}_{-1}^{-n} | x_0 = m) = (O \odot B^{(n-1)})\tilde{Q}, \quad (24)$$

and $A^{(1)} = OQ$ and $B^{(1)} = O\tilde{Q}$. In particular, for the given window size $N = 2n + 1$, we have:

$$A = A^{(n)}, \quad B = B^{(n)}, \quad C = O \text{Diag}(\pi). \quad (25)$$

The basic idea of *recovering* the minimal HMM realization θ^h (up to hidden state relabeling) is to first *recover* the factors A, B and C via tensor decomposition, and then extract the transition and observation probabilities from the factors. The minimal order condition is again reflected in the tensor *rank* factorization, as any HMM realization of lower order results in a tensor M of lower tensor rank, which is a contradiction.

e) *Identifiability*: The identifiability of the minimal HMM relies on the fact that the tensor rank decomposition indeed recovers the factor A, B, C defined in (20)–(22). Note that by definition, the column stochastic observation matrix O must have Kruskal rank greater than 2, otherwise there exist two identical columns in O , and the corresponding two hidden states can be merged to give an equivalent HMM realization of smaller order.

Lemma 3 (Uniqueness of tensor decomposition). *Given window size N , if the matrices $A, B \in \mathbb{R}^{d^n \times k}$ defined in (23)–(25) have full column rank k , then M can be uniquely decomposed into column stochastic matrices A, B, C as in (19) (up to common column permutation).*

In parallel with Theorem 1, the next theorem shows that the condition above is satisfied for a general position HMM process with sufficiently large window size N .

Theorem 3 (Choice of N for HMM realization). *Consider $\Theta_{(d,k)}^h$, the class of all HMMs with output alphabet size d and order k . There exists a measure zero set $\mathcal{E} \in \Theta_{(d,k)}^h$ such that for all output processes generated by HMMs in the set $\Theta_{(d,k)}^h \setminus \mathcal{E}$, the minimal quasi-HMM realization can be computed based on the joint probabilities in $\mathcal{P}^{(N)}$, if window size $N = 2n + 1$ for some n such that:*

$$n > 8 \lceil \log_d(k) \rceil. \quad (26)$$

f) Algorithms: The matrices A, B and C , defined in (23)–(25), are polynomial functions of the parameters Q and O of the minimal HMM realization. The following theorem exploits the recursive structure of these polynomials to recover the parameters Q and O if the factors A, B, C are given.

Theorem 4 (Recovering Q and O from A, B, C). *Given the matrix C , one can obtain the observation matrix by:*

$$O_{[:,i]} = C_{[:,i]} / (\mathbf{e}^\top C_{[:,i]}), \quad \forall i \in [k]. \quad (27)$$

Given the matrix $A \in \mathbb{R}^{d^n \times k}$, we first scale each of the column similar to (27) so that each column is stochastic, and corresponds to the conditional probabilities $\mathbb{P}(\mathbf{y}_1^n | x_0)$ as shown in (20). We marginalize the conditional distribution to get $A^{(1)} = \mathbb{P}(y_1 | x_0) \in \mathbb{R}^{d \times k}$ and $A^{(n-1)} = \mathbb{P}(\mathbf{y}_1^{n-1} | x_0) \in \mathbb{R}^{d^{n-1} \times k}$.

(1) If A has full column rank k ([1]):

$$Q = \left(O \odot A^{(n-1)} \right)^\dagger A. \quad (28)$$

(2) If C has full column rank k :

$$Q = O^\dagger A^{(1)}. \quad (29)$$

where $(X)^\dagger = (X^\top X)^{-1} X^\top$ denotes the pseudo-inverse of a matrix X .

In the proof of Theorem 3, we show that for general position HMMs with sufficiently large window size, the matrices A and B achieve full column rank k . When this holds, Algorithm 3 computes the unique tensor decomposition to recover the factors A, B, C . Theorem 4 (1) applies to recover Q and O from the factors.

However, if the transition matrix Q of the minimal HMM realization does not have full rank, and no matter how large the window size is, the matrix A never achieves full rank. Note that these HMMs are degenerate cases belonging to the measure zero set in Theorem 3, and Algorithm 3 is not applicable for decomposing the tensor M . However, it is still possible to apply Algorithm 4. Note that a necessary condition for it to work is that $d \geq k$ and the observation matrix is of full column rank.

Let $\Theta_{(d,k,r)}^h$ denote the model class of HMMs with output alphabet d and order k , for $d \geq k$ and the transition matrix Q has rank $r < k$. Note that $\Theta_{(d,k,r)}^h$ is a subset of the measure zero set \mathcal{E} in Theorem 3. The following theorem shows that if Algorithm 4 runs correctly for a random instance in this subset, then the algorithm works for almost all HMMs in this subset.

Theorem 5 (Correctness of Algorithm 4). *Given d, k and r and consider the set $\Theta_{(d,k,r)}^h$. Let A, B, C be defined as in (23)–(25) for $n = 1$, and let $M = A \otimes B \otimes C$. If Algorithm 2 returns “yes”, then there exists a measure zero set $\mathcal{E} \in \Theta_{(d,k,r)}^h$, such that Algorithm 4 returns the tensor decomposition $M = A \otimes B \otimes C$ for all HMMs in the set $\Theta_{(d,k,r)}^h \setminus \mathcal{E}$. Moreover, if the latter is true, Algorithm 2 returns “yes” with probability 1.*

For this class of degenerate HMMs, Theorem 4 (2) applies to recover Q and O .

Note that for both the general position case and this degenerate case, the computation complexity to recover the parameters of the minimal HMM realization are polynomial in both d and k , and this is an immediate result of the log upper bound of the window size.

V. CONCLUSION

In this paper, we discussed two realization problems. We show that for output processes generated by HMMs in general position, both learning the minimal quasi-HMM realization and learning the real minimal HMM realization are easy— in the sense that there exist efficient algorithms to compute the minimal realizations with running time and sample complexity both polynomial in the relevant parameters of the problem.

Algorithm 2 Check Condition

- 1) Randomly choose an HMM from $\theta^h \in \Theta_{(d,k,r)}^h$.
 - 2) Construct matrices A, B, C with (Q, O) as defined in (23)–(25) for $n = 1$, namely $A = OQ$, $B = O\tilde{Q}$, and $C = O\text{Diag}(\pi)$.
 - 3) Let $M = A \otimes B \otimes C$. Run Algorithm 4 with the input M .
 - 4) Return “yes” if the algorithm returns A, B, C uniquely up to a common column permutation, and “no” otherwise.
-

REFERENCES

- [1] Elizabeth S Allman, Catherine Matias, and John A Rhodes. Identifiability of parameters in latent structure models with many observed variables. *The Annals of Statistics*, pages 3099–3132, 2009.
- [2] Anima Anandkumar, Rong Ge, Daniel Hsu, Sham M Kakade, and Matus Telgarsky. Tensor decompositions for learning latent variable models. *arXiv preprint arXiv:1210.7559*, 2012.
- [3] Brian DO Anderson. The realization problem for hidden markov models. *Mathematics of Control, Signals and Systems*, 12(1):80–120, 1999.
- [4] Raphael Bailly. Quadratic weighted automata: Spectral algorithm and likelihood maximization. *Journal of Machine Learning Research*, 20:147–162, 2011.
- [5] Borja Balle, Xavier Carreras, Franco M Luque, and Ariadna Quattoni. Spectral learning of weighted automata. *Machine Learning*, pages 1–31, 2013.
- [6] Aditya Bhaskara, Moses Charikar, and Aravindan Vijayaraghavan. Uniqueness of tensor decompositions with applications to polynomial identifiability. *arXiv preprint arXiv:1304.8087*, 2013.
- [7] Lieven De Lathauwer. A link between the canonical decomposition in multilinear algebra and simultaneous matrix diagonalization. *SIAM Journal on Matrix Analysis and Applications*, 28(3):642–666, 2006.
- [8] Lieven De Lathauwer, Joséphine Castaing, and J Cardoso. Fourth-order cumulant-based blind identification of underdetermined mixtures. *Signal Processing, IEEE Transactions on*, 55(6):2965–2973, 2007.
- [9] Phillip Griffiths and Joseph Harris. *Principles of algebraic geometry*. John Wiley & Sons, 2014.
- [10] Daniel Hsu, Sham M Kakade, and Tong Zhang. A spectral algorithm for learning hidden markov models. *Journal of Computer and System Sciences*, 78(5):1460–1480, 2012.
- [11] Hisashi Ito, S-I Amari, and Kingo Kobayashi. Identifiability of hidden markov information sources and their minimum degrees of freedom. *Information Theory, IEEE Transactions on*, 38(2):324–333, 1992.
- [12] Tao Jiang and Nikos D Sidiropoulos. Kruskal’s permutation lemma and the identification of candecomp/parafac and bilinear models with constant modulus constraints. *Signal Processing, IEEE Transactions on*, 52(9):2625–2636, 2004.
- [13] Michael Kearns, Yishay Mansour, Dana Ron, Ronitt Rubinfeld, Robert E Schapire, and Linda Sellie. On the learnability of discrete distributions. In *Proceedings of the twenty-sixth annual ACM symposium on Theory of computing*, pages 273–282. ACM, 1994.
- [14] Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.
- [15] Joseph B Kruskal. Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics. *Linear algebra and its applications*, 18(2):95–138, 1977.
- [16] SE Leurgans, RT Ross, and RB Abel. A decomposition for three-way arrays. *SIAM Journal on Matrix Analysis and Applications*, 14(4):1064–1083, 1993.
- [17] Elchanan Mossel and Sébastien Roch. Learning nonsingular phylogenies and hidden markov models. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 366–375. ACM, 2005.
- [18] Nicholas D Sidiropoulos and Rasmus Bro. On the uniqueness of multilinear decomposition of n-way arrays. *Journal of chemometrics*, 14(3):229–239, 2000.
- [19] Eduardo D Sontag. On some questions of rationality and decidability. *Journal of Computer and System Sciences*, 11(3):375–381, 1975.
- [20] Sebastiaan A Terwijn. On the learnability of hidden markov models. In *Grammatical Inference: Algorithms and Applications*, pages 261–268. Springer, 2002.
- [21] Mathukumalli Vidyasagar. The complete realization problem for hidden markov models: a survey and some new results. *Mathematics of Control, Signals, and Systems*, 23(1-3):1–65, 2011.

APPENDIX A

TENSOR DECOMPOSITION ALGORITHMS

For completeness, we list two standard tensor decomposition algorithms in this section.

APPENDIX B

PROOFS

(Proof of Lemma 1)

If both E and F have full column rank k , by Sylvester inequality the rank of the matrix $H^{(0)}$ is also equal to k , the order of minimal quasi-HMM realization. Therefore, for the two matrices U and V obtained in Step 2 in Algorithm 3, there exists some full rank matrix $W \in \mathbb{R}^{k \times k}$ such that:

$$U = EW, \quad V^\top = W^{-1}F^\top.$$

Algorithm 3 Simultaneous diagonalization for 3rd order tensor decomposition [16]

Input: A 3rd order tensor $M \in \mathbb{R}^{d^n \times d^n \times d}$
Output: $k, A, B \in \mathbb{R}^{d^n \times k}, C \in \mathbb{R}^{d \times k}$

- 1) Randomly pick two unit norm vectors
- $\mathbf{v}_1, \mathbf{v}_2 \in \mathbb{R}^d$
- . Project
- M
- along the 3rd dimension to obtain two matrices:

$$\widetilde{M}_1 = M(I, I, \mathbf{v}_1), \quad \widetilde{M}_2 = M(I, I, \mathbf{v}_2).$$

- 2) Compute the eigen-decomposition of matrix
- $(\widetilde{M}_1 \widetilde{M}_2^{-1})$
- and
- $(\widetilde{M}_2 \widetilde{M}_1^{-1})$
- , and let the columns of matrix
- A
- and
- B
- be the eigenvectors of
- $(\widetilde{M}_1 \widetilde{M}_2^{-1})$
- and
- $(\widetilde{M}_2 \widetilde{M}_1^{-1})$
- , respectively.
-
- Scale the columns of
- A
- and
- B
- to be stochastic, and pair the eigenvectors in
- A
- and
- B
- corresponding to the reciprocal eigenvalues, namely:

$$\widetilde{M}_1 \widetilde{M}_2^{-1} = A \Lambda A^{-1}, \quad \widetilde{M}_2 \widetilde{M}_1^{-1} = B \Lambda^{-1} B^{-1}.$$

- 3) Let
- k
- be the number of non-zero eigenvalues.
-
- 4) Let
- $\overline{M}^{(3)} \in \mathbb{R}^{d^{2n} \times d}$
- be the 3rd dimension matricization of
- M
- . Set
- C
- to be:

$$C = \overline{M}^{(3)} ((A \odot B)^\dagger)^\top$$

Algorithm 4 FOABI for 3rd order tensor decomposition

Input: $M \in \mathbb{R}^{d \times d \times d}$
Output: k, A, B, C .

- 1) Let
- $\overline{M}^{(3)}$
- be the 3rd dimension matricization of
- M
- . Compute its SVD
- $\overline{M}^{(3)} = V_H D_H U_H^\top$
- .
-
- 2) Set
- k
- to be the number of non-zero singular values. Let
- $F = V_H D_H^{1/2}$
- , and
- $E = U_H D_H^{1/2}$
- .
-
- 3) Construct matrices
- $\{E^{(r)} \in \mathbb{R}^{d \times d} : r \in [k]\}$
- :

$$[E^{(r)}]_{i,j} = E_{(i-1)d+j,r}, \forall i, j \in [d], \forall r \in [k].$$

Construct the 4-th order tensors $\{P^{(r,s)} \in \mathbb{R}^{d \times d \times d \times d} : r, s \in [k]\}$:

$$\begin{aligned} [P^{(r,s)}]_{i_1, i_2, j_1, j_2} &= [E^{(r)}]_{i_1, j_1} [E^{(s)}]_{i_2, j_2} + [E^{(s)}]_{i_1, j_1} [E^{(r)}]_{i_2, j_2} \\ &\quad - [E^{(r)}]_{i_1, j_2} [E^{(s)}]_{i_2, j_1} - [E^{(s)}]_{i_1, j_2} [E^{(r)}]_{i_2, j_1}. \end{aligned}$$

- 4) Compute a basis
- $\{H^{(i)} : i \in [k]\}$
- of the
- k
- dimensional kernel of
- $\{P^{(r,s)} : r, s \in [k]\}$
- :

$$\sum_{r,s=1}^k H_{r,s}^{(i)} P^{(r,s)} = 0, \quad \text{s.t. } H_{r,s}^{(i)} = H_{s,r}^{(i)}, \forall r, s \in [k].$$

- 5) Find the unique
- $W \in \mathbb{R}^{k \times k}$
- that simultaneously diagonalizes the basis:

$$H^{(i)} = W \Lambda^{(i)} W^\top, \quad \forall i \in [k].$$

- 6) Let
- $C = F(W^{-1})^\top$
- and
- $A \odot B = EW$
- . Compute the rank one decomposition of each column of
- $A \odot B$
- , with proper normalization such that
- A
- and
- B
- are column stochastic.
-

Therefore, Step 3 returns

$$\widetilde{A}^{(j)} = W^{-1} E^\dagger E A^{(j)} F^\top (F^\top)^\dagger W = W^{-1} A^{(j)} W.$$

By the normalization constraint in Definition 1, we have

$$u^\top W = u^\top \sum_{j=1}^d A^{(j)} W = u^\top W \sum_{j=1}^d \widetilde{A}^{(j)}.$$

Moreover, since

$$U = \begin{bmatrix} u^\top(A^{(1)} \dots A^{(1)}) \\ u^\top(A^{(1)} \dots A^{(2)}) \\ \vdots \\ u^\top(A^{(d)} \dots A^{(d)}) \end{bmatrix} \quad W = u^\top W \begin{bmatrix} \tilde{A}^{(1)} \dots \tilde{A}^{(1)} \\ \tilde{A}^{(1)} \dots \tilde{A}^{(2)} \\ \vdots \\ \tilde{A}^{(d)} \dots \tilde{A}^{(d)} \end{bmatrix},$$

in Step 2 we obtain $\tilde{u}^\top = u^\top W$, and similarly, we can argue that $\tilde{v} = W^{-1}v$. Thus we conclude that the output $\tilde{\theta}^o = (k, \tilde{u}, \tilde{v}, \tilde{A}^{(j)} : j \in [d])$ is a valid minimal quasi-HMM realization of order k , and is equivalent to θ^o up to a linear transformation. \square

(Proof of Theorem 1)

Assume that the observed process has a minimal HMM realization θ^h of order k , i.e., $\theta^h \in \Theta_{(d,k)}^h$, and let θ^o denote the equivalent order k quasi-HMM as shown in (4)-(6). For window size $N = 2n + 1$, define the matrices E and F for θ^o as in (12) and (13) and note that:

$$\begin{aligned} E_{L(\mathbf{I}_1^n), i} &= [u^\top(A^{(l_n)} \dots A^{(l_1)})]_i \\ &= \mathbf{e}^\top \mathbb{P}(x_n, y_{n-1} = l_n | x_{n-1}) \dots \mathbb{P}(x_1, y_0 = l_1 | x_0 = i) \\ &= \mathbb{P}(\mathbf{y}_0^{n-1} = \mathbf{I}_1^n | x_0 = i), \end{aligned}$$

and similarly,

$$F_{L(\mathbf{I}_1^n), i} = [A^{(l_n)} \dots A^{(l_1)} \pi]_i = \mathbb{P}(\mathbf{y}_{-1}^{-n} = \mathbf{I}_1^n, x_0 = i).$$

Lemma 1 shows that a sufficient condition for the correctness of Algorithm 1 is that both E and F have full column rank k . In this proof, we show that when Q and O of the HMM $\theta^h \in \Theta_{(d,k)}^h$ are in general position, this rank condition is satisfied if the window size $N = 2n + 1$ satisfies (14).

Note that the minors of E and F are polynomials in the elements of Q and O , thus it defines an algebraic set in the parameter space by setting all the minors to zero to make E and F to be rank deficient. By basic algebraic geometry [9], the algebraic set either occupies the entire Zariski closure or is a low-dimensional manifold of Lebesgue measure zero. In particular, the Zariski closure of $\Theta_{(d,k)}^h$, defined to be the smallest algebraic set containing $\Theta_{(d,k)}^h$, is given by $\bar{\Theta}_{(d,k)}^h := \{O \in \mathbb{R}^{d \times k}, Q \in \mathbb{R}^{k \times k} : \mathbf{e}^\top O = \mathbf{e}^\top, \mathbf{e}^\top Q = \mathbf{e}^\top\}$ (note that the element-wise non-negativity constraints can be omitted when considering the Zariski closure). Therefore, it is enough to show that for some specific choice of Q and O in $\bar{\Theta}_{(d,k)}^h$, the matrices E and F achieve full column rank k . Moreover to construct an instance, we can further ignore the stochastic constraints, as scaling does not affect the independence property of the columns in E and F .

We fix the transition matrix Q to be the state shifting matrix as below:

$$Q_{i-1, i} = 1, \text{ for } 2 \leq i \leq k, \text{ and } \quad Q_{k, 1} = 1, \quad (30)$$

Note that with this choice of Q , $\pi = \frac{1}{k}\mathbf{e}$, and $\tilde{Q} = Q^\top$. Due to the symmetry of the forward and backward transitions, we can focus on showing that E has full column rank and the same argument applies to F .

We randomize the observation matrix O and let the columns be independent random variables uniformly distributed on the d -dimensional sphere. In order to show that there exists a construction of (Q, O) such that E has full column rank, it suffices to show that E achieves full column rank with positive probability over the randomness of O . We apply Gershgorin's theorem to prove that the columns of E are incoherent.

Note that for the shifting matrix Q , we have:

$$E_{[:, i]} = O_{[:, i]} \odot \dots \odot O_{[:, i+n-1]}.$$

Since we have $d \geq 2$ and $n < k$, for notational convenience, we slightly abuse notation to write the j -th column of O as $O_{[:, j]}$, while for $k < j \leq 2k$, it actually refers to the $(j - k)$ -th column of O .

Define matrix $X \in \mathbb{R}^{k \times k}$ to be:

$$X_{i, j} = E_{[:, i]}^\top E_{[:, j]} = \prod_{m=0}^{n-1} (O_{[:, i+m]}^\top O_{[:, j+m]}), \quad \forall i, j \in [k].$$

By the assumption that the columns of O are uniformly distributed on the d -dimensional sphere, we have $X_{i,i} = 1$, for all $i \in [k]$.

Fix some $\beta, \gamma = \beta^2 \in (0, 1)$. Suppose that, for any $i \neq j$,

$$\mathbb{P}\left(|X_{i,j}| < \frac{\beta}{k}\right) > 1 - \frac{\gamma}{k^2}. \quad (31)$$

Then apply union bound on j , we have for any i :

$$\begin{aligned} \mathbb{P}\left(\sum_{j \neq i}^k |X_{i,j}| < \beta\right) &\geq \mathbb{P}\left(\forall j \in [k], j \neq i, |X_{i,j}| < \frac{\beta}{k}\right) \\ &> 1 - \frac{\gamma}{k}. \end{aligned}$$

Again apply union bound on i , we have:

$$\mathbb{P}\left(\forall i \in [k], |X_{i,i}| - \sum_{j \neq i} |X_{i,j}| \geq 1 - \beta\right) > 1 - k \frac{\gamma}{k} = 1 - \gamma.$$

Apply Gershgorin's theorem, we have that with probability at least γ , the matrix $X = E^\top E$ is of full rank k , and the smallest singular value is at least $1 - \beta$. There must exist some instance of O such that this statement holds.

Next, we verify the statement in (31). Equivalently, we want to show that for $i \neq j$:

$$\begin{aligned} 1 - \frac{\gamma}{k^2} &< \mathbb{P}\left(\prod_{m=0}^{n-1} |O_{[:,i+m]}^\top O_{[:,j+m]}| < \frac{\beta}{k}\right) \\ &= \mathbb{P}\left(\sum_{m=0}^{n-1} \log(|O_{[:,i+m]}^\top O_{[:,j+m]}|) < -\log\left(\frac{k}{\beta}\right)\right) \\ &= \mathbb{P}\left(\sum_{m=0}^{n-1} \log\left(\frac{1}{|O_{[:,i+m]}^\top O_{[:,j+m]}|}\right) > \log\left(\frac{k}{\beta}\right)\right) \\ &= \mathbb{P}\left(\sum_{m=1}^n \log\left(\frac{1}{|v_m|}\right) > \log\left(\frac{k}{\beta}\right)\right) \end{aligned}$$

where v_m are i.i.d. random variables with the distribution as the projection of a uniform unit-norm vector in \mathbb{R}^d onto the first dimension. The last equality is due to the independence of the columns of O .

Define the indicator random variable s_m for $m \in [n]$:

$$s_m = \mathbf{1}\left[\log\left(\frac{1}{|v_m|}\right) < \frac{1}{c} \log(d)\right] = \mathbf{1}\left[|v_m| > \frac{1}{d^{1/c}}\right],$$

where we pick constant $c = 4$. Assume that $d \geq 2 + (8e)^2$ (as we really only care about the scaling), apply Johnson Lindenstrauss lemma (Lemma 9), setting u_1 to be v_m and t to be $1/d^{1/c}$, we have:

$$\mu = \mathbb{P}(s_m = 1) < \frac{4}{\sqrt{d-2}} e^{-\frac{d-2}{2d^{2/c}}} < \frac{1}{2e} e^{-\frac{d-2}{2d^{2/c}}}$$

Note that by definition:

$$\sum_{m=1}^n \log\left(\frac{1}{|v_m|}\right) > \sum_{m=1}^n \frac{1}{c} \log(d)(1 - s_m).$$

Therefore it suffices to show that

$$1 - \frac{\gamma}{k^2} < \mathbb{P}\left(\sum_{m=1}^n \frac{1}{c} \log(d)(1 - s_m) > \log\left(\frac{k}{\beta}\right)\right),$$

or equivalently,

$$\begin{aligned} \frac{\gamma}{k^2} &> \mathbb{P} \left(\sum_{m=1}^n s_m > n - c \frac{\log(k/\beta)}{\log(d)} \right) \\ &= \mathbb{P} \left(\sum_{m=1}^n s_m > \alpha c \frac{\log(k/\beta)}{\log(d)} \right) \end{aligned}$$

where we set $n = (1 + \alpha)c \log_d(k/\beta)$ for some $\alpha > 1$.

Apply the multiplicative Chernoff bound (Lemma 8), by setting $X_m = s_m$ for $m = 1, \dots, n$, and set $\delta n \mu = \alpha c \frac{\log(k/\beta)}{\log(d)}$, and $\frac{e}{\delta} = \frac{en\mu}{\alpha c \frac{\log(k/\beta)}{\log(d)}} = \frac{1+\alpha}{\alpha} e \mu < e^{-\sqrt{d}/2} < 1$, then we have

$$\mathbb{P} \left(\sum_{m=1}^n s_m > \alpha c \frac{\log(k/\beta)}{\log(d)} \right) < \left(\frac{1+\alpha}{\alpha} e \mu \right)^{\alpha c \frac{\log(k/\beta)}{\log(d)}}.$$

We want to show that the RHS is less than γ/k^2 . Taking log, this is equivalent to:

$$\alpha c \frac{\log(k/\beta)}{\log(d)} \log_d \left(\frac{\alpha}{(1+\alpha)e\mu} \right) > \frac{\log(k^2/\gamma)}{\log(d)}$$

Recall that we have $\gamma = \beta^2$, $\frac{1+\alpha}{\alpha} e \mu \leq e^{-\frac{d-2}{2d^{2/c}}}$, $c = 4$ the above inequality holds if we pick $\alpha = 4/c = 1$, as

$$\alpha c \log_d \left(\frac{\alpha}{(1+\alpha)e\mu} \right) \geq 4 \frac{\log(e^{\frac{\sqrt{d}}{2}})}{\log(d)} \geq 2 \frac{\sqrt{d}}{\log(d)} \geq 2.$$

Now we can conclude that (31) holds. □

(Proof of Theorem 2)

Recall that the output of Algorithm 1 is given by:

$$\begin{aligned} \hat{A}^{(j)} &= \hat{D}^{-1/2} \hat{U}_H^\top \hat{H}^{(j)} \hat{V}_H \hat{D}^{1/2}, \\ \hat{u} &= \hat{D}^{-1/2} \hat{U}_H^\top \mathbf{e}, \quad \hat{v} = \hat{D}^{-1/2} \hat{V}_H^\top \mathbf{e}, \end{aligned}$$

where \hat{U}_H and \hat{V}_H are the first k left and right singular vectors of $\hat{H}^{(0)}$, and the diagonal matrix \hat{D} has the first k singular values of $\hat{H}^{(0)}$ on its main diagonal. In order to bound the distance between $\hat{A}^{(j)}$ and $\tilde{A}^{(j)}$, \hat{u} and \tilde{u} , \hat{v} and \tilde{v} , we analyze the perturbation bound for each of the factor separately and apply Lemma 6 to bound the overall perturbation of the product form.

First, denote $E_j = \hat{H}^{(j)} - H^{(j)}$ for $j = 0, 1, \dots, d$. For any element in E_j we can be bound its norm using Hoeffding's inequality (Lemma 7): with probability at least $1 - 2e^{-2T\delta^2}$, the (i_1, i_2) -th element of E_j is bounded by: $\| [E_j]_{i_1, i_2} \| \leq \delta < 1$. Moreover, apply union bound to j and all elements in each E_j , with probability at least $1 - 2k^4 d^3 e^{-2T\delta^2}$, for all $j = 0, 1, \dots, d$, we have

$$\|E_j\|_F \leq \sqrt{k d^n} \delta < k^{1.5} d^{0.5} \delta,$$

where the last inequality is due to $d^n < k^2 d$.

Second, we apply the matrix perturbation bound (Lemma 5) to bound the distance of the singular vectors:

$$\|\hat{U}_H - U_H\| \leq \frac{\sqrt{2}\|E_0\|_F}{\sigma_k(H^{(0)})}, \quad \|\hat{V}_H - V_H\| \leq \frac{\sqrt{2}\|E_0\|_F}{\sigma_k(H^{(0)})}.$$

And we can apply Mirsky's theorem (Lemma 4) to bound the distance of the singular values:

$$\|\hat{D} - D\| \leq \|E_0\|_F.$$

Denote $\Delta_i = \sigma_i(\hat{H}^{(0)}) - \sigma_i(H^{(0)})$ and let $\sigma_i = \sigma_i(H^{(0)})$. Note that if $\|E_0\| \leq \sigma_k/2$, we have that for any $i = 1, \dots, k$, $|\Delta_i| \leq \|E_0\| \leq \sigma_i/2$, then

$$\begin{aligned} \left(\frac{1}{\sqrt{\sigma_i}} - \frac{1}{\sqrt{\sigma_i + \Delta_i}}\right)^2 &= \frac{1}{\sigma_i + \Delta_i} (\sqrt{1 + \Delta_i/\sigma_i} - 1)^2 \\ &\leq \frac{2}{\sigma_i} (\Delta_i/\sigma_i + 2 - 2\sqrt{1 + \Delta_i/\sigma_i}) \\ &\leq \frac{2}{\sigma_i} (3|\Delta_i|/\sigma_i) \\ &\leq \frac{6}{\sigma_k^2} |\Delta_i|, \end{aligned}$$

where the first inequality is due to $|\Delta_i| \leq \sigma_i/2$, and the second inequality is due to $\sqrt{1 + \Delta_i/\sigma_i} \geq 1 - |\Delta_i/\sigma_i|$. Therefore we have that

$$\|\hat{D}^{-1/2} - D^{-1/2}\| \leq \frac{\sqrt{6 \sum_{i=1}^k |\Delta_i|}}{\sigma_k} \leq \frac{\sqrt{6\sqrt{k} \|\hat{D} - D\|}}{\sigma_k}.$$

Finally, we apply Lemma 6 to bound the output perturbation. Note that $\|D^{-1/2}\| = 1/\sqrt{\sigma_k}$, $\|U_H\| = 1$, $\|V_H\| = 1$. Moreover note that the probabilities in each row of $H^{(j)}$ sum up to less than 1, therefore by Perron-Frobenius theorem we have $\|H^{(j)}\| \leq 1$. Therefore we have

$$\begin{aligned} &\|\hat{A}^{(j)} - \tilde{A}^{(j)}\| \\ &\leq 2^4 \left(\frac{2\sqrt{6}k^{1/2}\|E_0\|_F}{\sigma_k^{1.5}} + \frac{2\sqrt{2}\|E_0\|_F}{\sigma_k^2} + \frac{\|E_j\|}{\sigma_k} \right) \\ &\leq 2^4 \left(\frac{2\sqrt{6}k^{0.75}d^{0.25}\delta^{0.5}}{\sigma_k^{1.5}} + \frac{2\sqrt{2}kd^{0.5}\delta}{\sigma_k^2} + \frac{kd^{0.5}\delta}{\sigma_k} \right) \\ &\leq \frac{144kd^{0.5}}{\sigma_k^2} \delta^{0.5}, \end{aligned}$$

where the first inequality is due to $\|E_j\| \leq \|E_j\|_F$, and the second inequality is due to $\delta < 1$ and $\sigma_k \leq \sigma_1 \leq 1$.

Similarly we can bound $\|\hat{u} - \tilde{u}\|$ and $\|\hat{v} - \tilde{v}\|$ by:

$$\|\hat{u} - \tilde{u}\| \leq \|\hat{D}^{-1/2}\hat{U}_H^\top - D^{-1/2}U_H^\top\|\sqrt{d^n} \leq \frac{4k^{1.5}d}{\sigma_k^{1.5}}\delta^{0.5}.$$

In summary, if we want to achieve ϵ accuracy in the output, we need δ to be no larger than $\epsilon^2\sigma_k^4/(144k^3d^2)$. Set the failure probability to be $\eta = 2k^4d^3e^{-2T\delta^2}$, then number of sample sequences needed to estimate the empirical probabilities is given by:

$$T = 2 \frac{144^2k^6d^4}{\epsilon^4\sigma_k^8} \log \left(\frac{2k^4d^3}{\eta} \right).$$

□

(Proof of Theorem 3)

With exactly the same argument and constructional proof as for Theorem 1, we can show that for the window size $N = 2n + 1$ satisfies (26), the matrices A and B have full column rank. By Lemma 3 we have that the tensor decomposition of M is unique. Moreover, by the argument in Theorem 4 (1), we have that the model parameters Q, O can be uniquely recovered from the factors A, B, C . Thus in conclusion $\mathcal{P}^{(N)}$ is sufficient for finding the minimal HMM realization.

□

(Proof of Theorem 4)

By the uniqueness of tensor decomposition (up to column permutation and scaling) the columns of C are proportional to the columns of O (up to some hidden state permutation), and each column of O must satisfy the normalization constraint: $\mathbf{e}^\top O_{[:,i]} = 1, \forall i \in [k]$. The normalization in (27) recovers O from C .

Recall that

$$A = A^{(n)} = \left(O \odot A^{(n-1)} \right) Q.$$

Since the matrix A has full column rank k , the matrices $Q \in \mathbb{R}^{k \times k}$ and $(O \odot A^{(n-1)}) \in \mathbb{R}^{d^n \times k}$ both have full column rank k , as well as the pseudo-inverse of $(O \odot \tilde{A})$, therefore $Q = (O \odot A^{(n-1)})^\dagger A$.

By definition we have $A^{(1)} = OQ$, thus if O is of full column rank k , we can obtain $Q = O^\dagger A^{(1)}$. \square

(Proof of Theorem 5)

Denote the minimal order HMM realization by $\theta^h = (k, Q, O)$, and since $n = 1$, the matrices are given by:

$$A = OQ, \quad B = O\tilde{Q}, \quad C = O \text{Diag}(\pi).$$

Define two linear operators $I_{d^2 \times d^2} : \mathbb{R}^{d^2} \rightarrow \mathbb{R}^{d^2}$ and $P_{d^2 \times d^2} : \mathbb{R}^{d^2} \rightarrow \mathbb{R}^{d^2}$, such that for any matrix $X \in \mathbb{R}^{d \times d}$, $I_{d^2 \times d^2} \text{vec}(X) = \text{vec}(X)$ and $P_{d^2 \times d^2} \text{vec}(X) = \text{vec}(X^\top)$. Moreover, define matrix $R \in \mathbb{R}^{d^2 \times d^2}$ and $G \in \mathbb{R}^{d^4 \times d^2}$ to be:

$$R = I_{d^2 \times d^2} - P_{d^2 \times d^2}, \quad G = R \odot R.$$

Note that the kernel of $(I_{d^2 \times d^2} - P_{d^2 \times d^2})$ is the space of symmetric matrices, thus R is of rank $d^2 - d(d+1)/2 = d(d-1)/2$, and G is of rank $d^2(d-1)^2/4$. Define matrix $G^\perp \in \mathbb{R}^{d^4 \times (d^4 - \frac{d^2(d-1)^2}{4})}$ such that its columns are orthogonal to the columns of G .

According to [7], [8], [12], there are two deterministic conditions for Algorithm 4 to correctly recover the factors A, B, C from the rank k tensor M :

- 1) Both $A \odot B$ and C have full column rank k .
- 2) Define $T \in \mathbb{R}^{d^4 \times (m + (k-1)k/2)}$ to be:

$$T = \left[G_{[:,i]}^\perp : 1 \leq i \leq d^4 - \frac{d^2(d-1)^2}{4}, \right. \\ \left. A_{[:,k_1]} \odot A_{[:,k_2]} \odot B_{[:,k_1]} \odot B_{[:,k_2]} : 1 \leq k_1 < k_2 \leq k \right].$$

The columns of T are linear independent.

Parameterize the rank r transition matrix by $Q = UV^\top$ for some matrices $U, V \in \mathbb{R}^{k \times r}$. Define the parameter space \mathcal{Q} :

$$\mathcal{Q} = \{Q \in \mathbb{R}^{k \times k} : Q = UV^\top, U, V \in \mathbb{R}^{k \times r}, \mathbf{e}^\top Q = \mathbf{e}^\top\}$$

Note that by construction, the minors of $A \odot B$ and T are nonzero polynomials in the elements of the parameters U, V and O , in order to show that the two deterministic rank conditions are satisfied for almost all instances in the class $\Theta_{(d,k,r)}^h$, it is enough to construct an instance in the model class that satisfies the two conditions (by the random check in Algorithm 2). Moreover, if it is true, then with probability one, the two conditions are satisfied for a randomly chosen instance in the model class. \square

APPENDIX C AUXILIARY LEMMAS

(Matrix perturbation bounds)

Since the algorithms we have examined are all based on different forms of matrix decomposition. Characterizing the sample complexity boils down to analyzing the stability of the matrix decompositions. Here we review some well-known matrix perturbation bounds and prove some corollaries.

Given a matrix $\tilde{A} = A + E$ where E is a small perturbation, the following results bound the deviation of the singular vectors and singular values.

Lemma 4 (Mirsky's theorem). *Given matrices $A, E \in \mathbb{R}^{m \times n}$, with $m \geq n$, then*

$$\sqrt{\sum_{i=1}^n (\sigma_i(A + E) - \sigma_i(A))^2} \leq \|E\|_F.$$

Lemma 5. Given matrices $A, E \in \mathbb{R}^{m \times n}$, with $m \geq n$. Suppose that the matrix A has full column rank and $\sigma_k(A) > 0$. Let $A = USV^\top$ be the singular value decomposition of A , and let \hat{U} and \hat{V} denote the first k left and right singular vectors of \hat{A} , let \hat{S} be the diagonal matrix with the first k singular values of \hat{A} . We have:

$$\|\hat{U} - \tilde{U}\| \leq \frac{\sqrt{2}\|E\|_F}{\sigma_k(A)}, \quad \|\hat{V} - V\| \leq \frac{\sqrt{2}\|E\|_F}{\sigma_k(A)}.$$

This is an immediate corollary of Wedin's theorem.

Lemma 6. Consider a product of matrices $A_1 \cdots A_k$, and consider any sub-multiplicative norm on matrix $\|\cdot\|$. Given $\hat{A}_1, \dots, \hat{A}_k$ and assume that $\|\hat{A}_i - A_i\| \leq \|A_i\|$, then we have:

$$\|\hat{A}_1 \cdots \hat{A}_k - A_1 \cdots A_k\| \leq 2^{k-1} \prod_{i=1}^k \|A_i\| \sum_{i=1}^k \frac{\|\hat{A}_i - A_i\|}{\|A_i\|}.$$

(Concentration bounds)

Lemma 7 (Hoeffding's inequality). Let X_1, \dots, X_n be independent random variables. Assume that X_i 's are bounded almost surely, namely $\Pr[X_i \in [a_i, b_i]] = 1$. Define the empirical mean of these variables $\bar{X} = (X_1 + \dots + X_n)/n$. We have

$$\Pr[|\bar{X} - \mathbb{E}[\bar{X}]| \geq t] \leq \exp\left(-\frac{2n^2 t}{\sum_{i=1}^n (b_i - a_i)^2}\right).$$

Lemma 8 (Multiplicative Chernoff bound). Suppose X_1, \dots, X_n are independent random variables with Bernoulli distribution, and $\mathbb{P}(X_i = 1) = \mu$. Then for any $\delta > 1$:

$$\mathbb{P}\left(\sum_{i=1}^n X_i > \delta n \mu\right) < \left(\frac{e}{\delta}\right)^{\delta n \mu}.$$

Lemma 9 (High dimensional sphere projection (Johnson Lindenstrauss lemma)). Let the random vector $\mathbf{u} \in \mathbb{R}^d$ be uniformly distributed on the surface of the d -dimensional unit sphere, i.e. uniform distribution in the set: $\left\{\sum_{i=1}^d u_i^2 = 1\right\}$. Denote its projection onto the first dimension to be $|u_1|$. We have:

$$\mathbb{P}(|u_1| > t) < \frac{4}{\sqrt{d-2}} e^{-\frac{d-2}{2} t^2}.$$