Minimal Realization Problems for Hidden Markov Models

Qingqing Huang†, Rong Ge†, Sham Kakade†, Munther Dahleh†

Abstract—Consider a stationary discrete random process with alphabet size $d$, which is assumed to be the output process of an unknown stationary Hidden Markov Model (HMM). Given the joint probabilities of finite length strings of the process, we are interested in finding a finite state generative model to describe the entire process. In particular, we focus on two classes of models: HMMs and quasi-HMMs, which is a strictly larger class of models containing HMMs. In the main theorem, we show that if the random process is generated by an HMM of order less or equal than $k$, and whose transition and observation probability matrix are in general position, namely almost everywhere on the parameter space, both the minimal quasi-HMM realization and the minimal HMM realization can be efficiently computed based on the joint probabilities of all the length $N$ strings, for $N > 4[\log_d(k)] + 1$. In this paper, we also aim to compare and connect the two lines of literature: realization theory of HMMs, and the recent development in learning latent variable models with tensor decomposition techniques.

I. INTRODUCTION

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 order of an HMM is defined to be the dimension of hidden state space, and it is an indicator of the model complexity. Compared to an arbitrary random process, the Markovian property of the hidden state evolution potentially gives more structures to the output process and thus reduces its representational complexity. In this work, we try to address the 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 in an efficient way?

Consider a discrete random process $\{y_t : t \in \mathbb{Z}\}$, which assumes values in a finite alphabet $[d] \equiv \{1, \cdots, d\}$. Let $y_N = (y_1, \cdots, y_N)$ denote a string of length $N$, which assumes values in the $N$-ary Cartesian product $[d]^N$. We assume that the process is the output of a stationary HMM of a finite but unknown order.

Note that a stationary random process is fully characterized by the joint probabilities of strings of any length, which we denote by an infinitely countable table:

$$\mathcal{P}^{(\infty)} = \left\{ P(y_1 = l_1, \cdots, y_N = l_N) : \forall l_i \in [d]^N, \forall N \in \mathbb{Z} \right\}.$$ 

The goal is to exploit the fact that it is the output process of a finite order HMM to reduce the representational complexity. In particular, we address the following issues:

1) (Informational complexity) Suppose that we are given the joint probabilities of all the length $N$ strings, denoted by:

$$\mathcal{P}^{(N)} = \left\{ P(y_1 = l_1, \cdots, y_N = l_N) : \forall l_i \in [d]^N \right\}.$$

If the underlying HMM has minimal order $k$, how large $N$ needs to be such that the joint probabilities in $\mathcal{P}^{(N)}$ are sufficient to pin down $\mathcal{P}^{(\infty)}$?

2) (Computational complexity) For sufficiently large $N$, and given the exact probabilities in $\mathcal{P}^{(N)}$, can we find a finite state system that can produce a stationary random process with the same joint probabilities in $\mathcal{P}^{(\infty)}$ (realize the observed process) in a computationally efficient way?

3) (Statistical complexity) In practice, we only have access to a finite number of sample sequences of the random process, based on which we can compute empirical estimates of the joint probabilities in $\mathcal{P}^{(N)}$. Are the realization algorithms robust to estimation noise of the input $\mathcal{P}^{(N)}$?

These are long standing questions, and there are several lines of work within different communities attempting 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 \cite{15, 19}. However, a more practical question is can we efficiently solve the realization / learning problem for most of HMMs? In this work, we focus on generic analysis and show that for almost all HMMs, i.e., excluding those whose transition and observation matrix are in a measure...
zero set in the parameter space, the output processes can be efficiently realized with finite state models. In particular, the algorithms have computational complexity and sample complexity that are both polynomial in the alphabet size $d$ and the system order $k$.

In particular, we shall address the above realization problems using spectral method based algorithms. Spectral method acquired the name by relying on variations of singular value decompositions at its core. Different forms of spectral method have appeared as powerful techniques for solving problems in signal processing [10], linear system identification (subspace identification) [23], and learning graphical models with latent structures [2]. For HMM realization problems, the key idea of spectral method 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. The rank of the factorization immediately reveals the minimal order of the minimal models, and the model parameters can then be extracted from the factors of the decomposition.

In Section III, we consider the problem of realizing the process with quasi-HMM models. Quasi-HMMs are associated with different names in different literature, including finite state regular automata [4], [5], regular quasi realization [19], [25], and operator models [13], [19]. In our discussion, we mostly follow the definitions and terminologies in [25]. In Theorem 1, we show that, for an output process generated by an HMM with minimal order $k$ and output alphabet size $d$, if the HMM is in general position, namely excluding a measure zero set in the parameter space of the joint probabilities in $\mathcal{P}(N)$ with the window size $N$ in the order of $O(\log_d(k))$ are sufficient for pinning down the entire random process $\mathcal{P}(\infty)$. Moreover, Algorithm 1 computes the minimal quasi-HMM realization with running time and sample complexity both polynomial in the relevant parameters, including $k$ and $d$. In the past few years, tensor decompositions proliferated for learning latent variable models [1], [2]. This type of spectral method circumvents the positivity and stochasticity constraints of the model parameters, and relies on the uniqueness of tensor decomposition to identify the underlying model. Tensor decomposition also provides a new way of looking at the HMM minimal realization problems. In [2], the authors show how to efficiently identify the underlying HMM when $d > k$, and the transition probability matrix of the underlying HMM is assumed to be of full rank. The proposed algorithm constructs a 3-rd order tensor with the joint probabilities of window size $N = 3$. Tensor decomposition based algorithms are also discussed in [1], [7]. However, for the general case where $d$ can be possibly smaller than $k$, it has not been showed rigorously that the algorithm has polynomial running time. In [1], the authors examined the generic identifiability conditions of HMM, and constructed an example to show that the window size $N = 2n + 1$ needs to satisfy $\left(\frac{n + d - 1}{d - 1}\right) \geq k$. In the case where $d$ is much smaller than $k$, the required $n$ is in the order of $O(k^{1/d})$ and is polynomial in $k$. In [7], the provided bound on the window size $N$ is in the order of $O(k/d)$, which is again polynomial in $k$. Therefore, since the dimensions of the tensors considered in these works are exponential in $n$, with such window size, the running time of the algorithms are exponential in $k$.

In Section IV, we examine HMM realization problems for the general case where the alphabet size $d$ can be greater than the order $k$. We first propose a two-step realization approach, and analyze the identifiability issue of the two steps. Then, similar to the results for minimal quasi-HMM realizations, we show that excluding a measure zero set in the parameter space of HMMs, the window size $N$ only needs to be in the order of $O(\log_d(k))$ for the HMM to be efficiently identified from the joint probabilities in $\mathcal{P}(N)$.

II. MINIMAL REALIZATION PROBLEM FORMULATION

In this section, we first review the basics of HMMs, and then we formally introduce the two realization problems: realizing the process with quasi Hidden Markov Models and with real Hidden Markov models.

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 denote the state space of the hidden states by $[k]$, and the number of hidden states $k$ is defined to be the order of the HMM.

A stationary HMM with alphabet size $d$ and order $k$ is parameterized by a pair of column stochastic matrices: the state transition matrix $Q \in \mathbb{R}^{d \times k}$, and the observation matrix $O \in \mathbb{R}^{d \times k}$ which satisfy $e^\top Q = e^\top$, $e^\top O = e^\top$, where $e$ 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 the $k$-dimensional vector $\pi$ denote the stationary state distribution, i.e., $\pi_i = P[x_t = i]$ and $Q \pi = \pi$. Without loss of generality, we assume that for the stationary process, $\pi_i > 0$ for all $i \in [k]$. We also define the backward transition matrix $\tilde{Q} \in \mathbb{R}^{k \times k}$:

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

It is easy to see that the matrix $\tilde{Q}$ is related to $Q$ in the following way:

$$
\tilde{Q} = \text{Diag}(\pi) Q^\top \text{Diag}(\pi)^{-1},
$$

Conditioned on the hidden state taking value $i$, the probability of observing letter $j$ is given by:

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

We call two HMMs equivalent if the output processes are statistically indistinguishable. Note that an obviously equivalent class of HMM is generated by permuting the hidden states. In other words, if the columns of the transition and observation matrices of two HMMs only differ by a common permutation, the two HMMs are equivalent.

Moreover, we will denote the class of all HMMs with output alphabet size $d$ and order $k$ by $\Theta_{(d,k)}^n$.

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

### B. Problem formulations

In practice, we can only estimate the joint probabilities of finite length strings, or other statistics of the process, from the sample output sequences, and try to estimate the underlying HMM based on the estimated probabilities. The realization problem first neglects the estimation error and assumes that we are given $P^{(N)}$, the exact joint probabilities of all length-$N$ strings, for a fixed window size $N$. The goal is to find a finite state model of the minimal order to “realize” the entire output process $P^{(\infty)}$, i.e., characterizing the joint probabilities of strings of any length.

Next, we introduce two finite state model classes, both of which can be used to realize the random process generated by a stationary HMM.

**Definition 1** (Quasi-HMM realization). Let $\theta^o$ be a tuple: $\theta^o = (d, k, u, v, A^{(j)} : j \in [d])$. We call $\theta^o$ a valid quasi-HMM realization of order $k$ for a stationary random process $\{y_t : t \in \mathbb{Z}\}$ if the three conditions hold: ($\forall y_1^N \in [d]^N$, $\forall N \in \mathbb{Z}$)

$$
\begin{align*}
P(y_1^N = 1_1^N) &= u^\top A^{(l_1)} A^{(l_2)} \ldots A^{(l_N)} v, \\
u^\top \left( \sum_{j=1}^d A^{(j)} \right) &= u^\top, \\
\left( \sum_{j=1}^d A^{(j)} \right) v &= v.
\end{align*}
$$

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

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

Note that by definition two equivalent quasi-HMMs correspond to the same stationary random process.

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

**Definition 3** (HMM realization). Let $\theta^h$ be a tuple: $\theta^h = (d, k, O \in \mathbb{R}^{d \times k}, Q \in \mathbb{R}^{k \times k})$. We call $\theta^h$ an HMM realization 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 the random process.

Note that HMMs form a subset of quasi-HMMs, since given any valid HMM realization $\theta^h = (d, k, O, Q)$, one can construct the following valid quasi-HMM realization $\theta^o = (d, k, u, v, A^{(j)} : j \in [d])$:

$$
\begin{align*}
u &= e, \\
v &= \pi, \\
A^{(j)} &= Q \text{Diag}(O_{\{j\}}), \quad \forall j \in [d].
\end{align*}
$$

where $\pi = Q \pi^t$ is the right eigenvector with an eigenvalue of one, and $\pi_i > 0$ for all $i \in [k]$. It is easy to verify that the constraints (3)–(5) are satisfied.

The minimal HMM realization problem is formally stated below: Assume the random process is the output of an HMM. How large $N$ needs to be so that we can efficiently construct a minimal HMM realization from $\mathcal{P}^{(N)}$ for the random process?

**III. Minimal Quasi-HMM Realization**

In this section, we address the minimal quasi-HMM realization problem. We first review the widely used
algorithm and examine the conditions that guarantee the correctness; then we show for HMMs in general position, in order to satisfy those conditions, the window size $N$ only needs to be in the order of $O(\log_2(k))$; we also give an example of hard case (degenerate) which needs $N$ to be as large as $k$; finally we show that the algorithm is stable and robust to input perturbation.

A. Algorithm

In this section, we first provide the intuition of the algorithm for finding the minimal quasi-HMM realization, and state the deterministic conditions for its correctness.

Given $P(N)$, the probabilities of all the strings of length $N = 2n + 1$, i.e., $P\left(y_{-n:n} = l_{-n:n}\right)$, for all $l_{-n:n} \in [d]^N$, we form the $d^n \times d^n$ matrices $H^{(0)}$ and $H^{(j)}$ for all $j \in [d]$ as below:

$$
\begin{align*}
\left[H^{(0)}\right]_{L(l_1^n),L(l_{-1}^n)} &= P\left(y_{-n:n} = l_{-n:n}, y_0^{-1} = l_1^n\right), \\
\left[H^{(j)}\right]_{L(l_1^n),L(l_{-1}^n)} &= P\left(y_{-1:n} = l_{-1:n}, y_0 = j, y_1^n = l_1^n\right),
\end{align*}
$$

where $l_1^n = (l_1,\ldots,l_n) \in [d]^n$ denotes the length $n$ string corresponding to the future $n$ time slots, and $l_{-1}^n = (l_{-1},l_{-2},\ldots,l_{-n}) \in [d]^n$ denotes the length $n$ string corresponding to past $n$ time slots. Note that the first dimension of the matrix corresponds to the “future” and the second dimension corresponds to the “past”, which are independent conditioned on the hidden “current” state due to the underlying Hidden Markov Model that generates the process. This is the key property the algorithm exploits to find the minimal realization.

Assume $\theta^o = (d,k,u,v,A^{(j)} : j \in [d])$ is a minimal order quasi-HMM model for the random process. By definition as shown in [5], the factorization of the joint probabilities in terms of the $A^{(j)}$’s allows us to write the matrix $H^{(0)}$ into the product of two matrices $E,F \in \mathbb{R}^{d^n \times k}$ as below:

$$
H^{(0)} = EF^T,
$$

and similarly, matrix $H^{(j)}$ for all $j$ can be written as:

$$
H^{(j)} = EA^{(j)}F^T,
$$

where the matrices $E,F$ are functions of the parameters of $\theta^o$. In particular, the $L(l_1^n)$-th row of $E$ and $F$ are given by: ($\forall l_1^n \in [d]^n$)

$$
\begin{align*}
E_{[L(l_1^n),:]} &= u^{\top}(A(l_{n}) \ldots A(l_1)), \\
F_{[L(l_1^n),:]} &= v^{\top}(A(l_{n}) \ldots A(l_1))^\top.
\end{align*}
$$

Note that if both $E$ and $F$ have full column rank $k$, then by Sylvester’s inequality the rank of $H^{(0)}$ is equal to $k$.

Algorithm 1 below takes $H^{(0)}$ and $H^{(j)}$’s as input, solves a matrix factorization problem, and returns a minimal order quasi-HMM realization that is equivalent to $\theta^o$ up to a linear transformation.

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

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

$$
H^{(0)} = U_H D_H V_H^\top.
$$

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

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

$$
\bar{u} = U^\top e, \quad \bar{v} = V^\top e.
$$

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

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

Algorithm 1 can be found in the literature in slightly different forms [3, 5]. The correctness of the algorithm crucially relies on that the matrix $H^{(0)}$ achieves the maximal rank, which is equal to the order of the minimal realization. We first state a necessary condition for the correctness of the algorithm.

**Lemma 1 (Correctness of Algorithm 1).** If the matrices $E,F$ defined in (11) and (12) have full column rank $k$, then Algorithm 1 returns a quasi-HMM realization $\bar{\theta}^o$ that is equivalent to $\theta^o$ up to linear transformation.

Note that $H^{(0)}$ is constructed with the probabilities of the length $N$ strings, and increasing the window size $N$ can potentially boost the rank of $H^{(0)}$. However, in previous works, people either assume that $N$ is large enough so that the maximal rank is achieved [5], or $N$ is gradually increased till the rank of $H^{(0)}$ does not increase any more [3]. To the best knowledge of the authors, there is no guidance on the window size selection to ensure the correctness of the algorithm, thus there is no guarantee on the computational complexity of the algorithm.

B. Generic analysis of information complexity

Since the dimension of the matrix $H^{(0)}$ is exponential in $N$, we desire a window size $N$ as small as possible while guaranteeing the full column rank of the matrices $E$ and $F$ defined in (11) and (12), and thus the correctness of Algorithm 1. In this part, we show that if the random process is generated by an order $k$ HMM in general position, then window size $N > 4\lceil \log_4(k) \rceil + 1$ is enough. In other words, $P(N)$ is sufficient for realizing the entire random process.

**Theorem 1 (Choice of $N$ for quasi-HMM realization).**
(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 the set $\Theta_{d,k}^h \setminus \mathcal{E}$, Algorithm 7 returns a minimal quasi-HMM realization, for window size $N = 2n + 1$ such that:

$$n > 2[\log_d(k)].$$

(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 $H^{(0)}$ constructed with $\mathcal{P}^{(N)}$ achieves the 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 7.

Note that the elements of matrices $E$ and $F$ are polynomials of the HMM parameters $Q$ and $O$, thus in order to show $E$ has full column rank, it suffices by constructing an instance of HMM and show that the columns of $E$ are incoherent for the choice of window size. In the construction, we pick a particular transition matrix $Q$ and randomize the observation matrix. The proof used the properties of random matrices, which are reviewed in Appendix B. The detailed proof is provided in Appendix A.

Also, a necessary condition is given by: $d^n > k$. 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 $n = [\log_d(k)]$ is sufficient. Even though in Theorem 1(a) the proved bound is $n > 2\log_d(k)$, we conjecture that in general, $n \geq \log_d(k)$ is enough.

C. Comment on the hard cases

Another body related to our work is the hardness results of learning HMMs in [15], [19], [22]. Those works studied the worst case scenarios and showed that learning the distribution of an HMM can be hard under cryptographic assumptions.

In Fig. 1 we adapt the hardness results to our setting and give an example of stationary HMM where the state diagram describes the transition and observation probabilities. In order the realizing the output process of this HMM, it is equivalent to learning the joint distribution of the process. It is easy to verify that the window size $N$ needs to be at least $T$, which is proportional to the order of the underlying HMM. This choice of window size directly results in informational and computational complexity that are exponential in the order of the HMM.

Note that this does not contradict our previous generic analysis, as this degenerate case belongs to the measure zero set in the parameter space of all HMMs.

We also point out that, for other HMMs in the measure zero set $\mathcal{E}$ that Algorithm 1 cannot handle, unlike the above example which can be reduced to the problem of learning noisy parity, they are not necessarily hard cases in the information theoretical sense. Those instances are only hard cases for the particular realization algorithm considered in this paper, and it is possible that some of them can be efficiently solved by other algorithms.

In particular, consider the degenerate HMM with the transition matrix $Q$ equal to the identity matrix, and with an arbitrary observation matrix $O$ and suppose that $d < k$. It is shown in [11] that $N$ needs to be at least in the order of $k^\frac{d}{2}$ such that the matrices $E$ and $F$ attain the full column rank, and therefore Algorithm 1 is exponential in $k$ for this case. However this degenerate HMM without mixing in the hidden states evolution is not fundamentally difficult.

D. Stability analysis

The realization algorithms we have discussed so far are based on the exact joint distributions of $y^N_1 = (y_1, \ldots, y_N)$ in $\mathcal{P}^{(N)}$. However, in practice, we only have access to estimated probabilities computed from finite sample data of the random process. How are the accuracy of the statistics affect the estimates on the parameters of the minimal realization? In this section, we shall analyze the robustness of the realization algorithms and derive the sample complexity bounds.

We show that in order to achieve $\epsilon$-accuracy in the parameters of the output quasi-HMM, the number of samples sequences needed is polynomial in all relevant parameters, including the order $k$.

For minimal quasi-HMM realization, the sample complexity bound for Algorithm 1 is given by the following theorem.

![State Diagram](image)
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 (9) and (10) with the empirical probabilities. Let \( N = 2n + 1 \), and \( n = 2\lceil \log_d(k) \rceil \). Let \( \hat{\vartheta}^j = (d, k, u_j, \tilde{v}_j, A^{(j)} : j \in [d]) \) and \( \tilde{\vartheta}^j = (d, k, \tilde{u}_j, \tilde{v}_j, A^{(j)} : j \in [d]) \) be the output of Algorithm 2 with the empirical probabilities and the exact probabilities for the input, respectively. Then, in order to achieve \( \epsilon \)-accuracy in the output with probability at least \( 1 - \eta \), namely:

\[
\| \tilde{u} - u \| \leq \epsilon, \quad \| \tilde{v} - v \| \leq \epsilon, \quad \| \tilde{A}^{(j)} - A^{(j)} \| \leq \epsilon, \quad \forall j,
\]

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

\[
T = Ck^6d^4 \frac{1}{\epsilon^4\sigma_k^6} \log \left( \frac{2kd^2}{\eta} \right),
\]

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

Since the algorithm relies on the singular value decomposition of the matrix \( H^{(0)} \) and other steps involve basic matrix inversions and multiplications, the proof of stability uses the spectral properties of matrix perturbation, and we review some standard results in Appendix B. The detailed proof is provided in Appendix A.

Remark 1. Note that the sample complexity is polynomial in \( k, d, 1/\epsilon, 1/\sigma_k \).

Theorem [7] showed that for HMMs in general position, \( n \geq 2\lceil \log_d(k) \rceil \) guarantees that \( \sigma_k > 0 \). Moreover, since \( \|H^{(0)}\| \leq 1 \) (by Perron-Frobenius), thus \( \sigma_k \geq |\det(H^{(0)})| \), where \( \det(H^{(0)}) \) is a polynomial of the model parameters \( Q \) and \( O \). Then we can apply the anti-concentration bound for polynomials in [8] to show that when the parameters \( Q \) and \( O \) are perturbed by adding small Gaussian noise, the smallest singular value of \( \sigma_k(E) \) can be bounded below with high probability.

However, a more rigorous way to show the stability of the algorithm in practice, and in particular lower bounding the condition number \( \sigma_k \), is to conduct the analysis in the smoothed analysis framework first proposed in [27], which is deferred to future work.

IV. MINIMAL HMM REALIZATION PROBLEM

Recall that an HMM realization can be easily converted to a valid quasi-HMM realization of the same order as shown in (6)–(8), yet it is difficult to construct an HMM if given an arbitrary quasi-HMM realization [3]. However, an HMM is more interpretable than a quasi-HMM especially for applications where the hidden states have physical meanings. In this section, we address the minimal HMM realization problem and connect the results to the previous sections. In particular, we show that for processes generated by HMMs in general position, solving the two problems have similar difficulty.

Unlike that the minimal quasi-HMM realizations are equivalent up to linear transformation, the minimal HMM realizations are equivalent only up to hidden states permutation. We apply tensor decomposition similar to that in [1], [7] to solve this problem. We first provide some background on tensor algebra, which are helpful for understanding the conditions that guarantee the identifiability of the minimal HMM realization. Then we state the sufficient conditions on the identifiability, and show when those conditions are satisfied for processes generated by HMMs in general position. Finally we show that two efficient tensor decomposition techniques can be applied in our setting to find the minimal HMM realization.

A. Preliminaries on tensor algebra

Tensor algebra has many similarities to but also many striking differences from matrix algebra, one of which is that, under very mild conditions, higher order tensor decomposition is unique up to scaling and column permutation of the factors, and this is the key property we exploit to uniquely identify the minimal HMM realization. We review some properties of 3-rd order tensors, and they can be immediately extended to tensors of higher orders. A more detailed introduction to tensor algebra can be found in [16] and the references therein.

One way to view a 3-rd order tensor \( X \in \mathbb{R}^{n_A \times n_B \times n_C} \) is that it defines a three-way array, with the multi-index \( (j_1, j_2, j_3), \forall j_1 \in [n_A], j_2 \in [n_B], j_3 \in [n_C] \). 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 decomposition extends singular value decomposition of matrices to higher order tensors, and writes a tensor into a sum of rank-1 tensors.

Definition 4 (Tensor rank decomposition). The rank decomposition of a 3-rd 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 \( k \):

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

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 3-rd 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: \( (\forall j_1 \in [m_A], j_2 \in [m_B], j_3 \in [m_C]) \)

\[ [X(V_1, V_2, V_3)]_{j_1,j_2,j_3} = \sum_{i_1 \in [n_A], i_2 \in [n_B], i_3 \in [n_C]} X_{i_1,i_2,i_3} [V_1]_{j_1,i_1} [V_2]_{j_2,i_2} [V_3]_{j_3,i_3}. \] (17)

An equivalent way to represent the multi-linear mapping better visualize the operator. Assume that the tensor admits a decomposition \( X = A \otimes B \otimes C \in \mathbb{R}^{n_A \times n_B \times n_C} \), then

\[ X(V_1, V_2, V_3) = (V_1 A) \otimes (V_2 B) \otimes (V_3 C), \] (18)

Note that \( X(V_1, V_2, V_3) \) is uniquely defined as in (17). \( X \) can have more than one form of decompositions, yet (18) defined for different decompositions are 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 \( X = A \otimes B \in \mathbb{R}^{n_A n_B \times k} \) is defined as follows:

\[ X_{(j_1-1)n_B+j_2,i} = A_{j_1,i} B_{j_2,i}, \quad \forall j_1 \in [n_A], j_2 \in [n_B], i \in [k], \] and each column of \( X \) is a rank-1 Khatri-Rao product.

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

\[ X_{j_3, (j_1-1)n_B+j_2} = X_{j_1,j_2,j_3}. \]

Moreover, if the tensor admits a decomposition \( X = A \otimes B \otimes C \), we can write the matricization in terms of the factors. For example, \( X^{(3)} = C(A \otimes B)^T \in \mathbb{R}^{n_C \times n_A n_B} \).

**B. Minimal HMM realization**

In this part, we explain how to apply tensor decomposition to solve the minimal HMM realization problem. For a fixed window size \( N = 2n + 1 \), given the exact joint probabilities in \( \mathcal{P}(N) \), similar to the construction of matrices \( H^{(0)} \) in (9), one can construct a 3-rd order tensor \( M \in \mathbb{R}^{d^n \times d^n \times d^n} \) as below:

\[ M_{L(1^n), L(1^n), i_0} = \mathbb{P}(y^*_n = 1^n_n), \quad \forall 1^n_n \in [d]_N. \] (19)

Suppose that there exists an order \( k \) minimal HMM realization \( \theta^h = (d, k, Q, O) \), then we can write the tensor \( M \) as a tensor product of matrices:

\[ M = A \otimes B \otimes C, \] (20)

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

\[ A_{L(1^n), m} = \mathbb{P}(y^*_1 = 1^n_1 | x_0 = m), \] (21)

\[ B_{L(1^n), m} = \mathbb{P}(y^-_1 = 1^n_1^- | x_0 = m), \] (22)

\[ C_{l,m} = \mathbb{P}(y_0 = l, x_0 = m). \] (23)

Thus \( A, B, C \) are column stochastic matrices. Moreover, \( A \) and \( B \) are recursive linear functions of the model parameters \( Q \) and \( O \) as below:

\[ A^{(n)} = \mathbb{P}(y^*_1 | x_0 = m) = (O \otimes A^{(n-1)}Q), \] (24)

\[ B^{(n)} = \mathbb{P}(y^-_1 | x_0 = m) = (O \otimes B^{(n-1)}Q). \] (25)

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

\[ A = A^{(n)}, \quad B = B^{(n)}, \quad C = O \text{Diag}(\pi). \] (26)

The basic idea of finding the minimal HMM realization \( \theta^h \) that is equivalent to \( \theta^h \) up to hidden state permutation, is to first obtain the factors \( A, B \) and \( C \) via tensor decomposition of \( M \), and then extract the transition and observation probabilities from the factor matrices.

**C. Sufficient conditions for identifiability**

The identifiability of the minimal HMM realization \( \theta^h \) is guaranteed by the uniqueness of the tensor decomposition in (20) and the invertibility of the mapping from \( Q \) and \( O \) to the factors \( A, B, C \) defined in (24)–(26).

In this part, we first state a set of sufficient conditions on the matrices \( A, B, C \) which guarantees that the factorization of \( M \) in (20) is indeed the unique tensor decomposition (up to common column permutation), and the rank of the tensor \( M \) is equal to the order of the HMM \( \theta^h \). Then we show that for a moderate choice of the window size \( N \), these conditions are satisfied.

First, we introduce the definition of Kruskal rank, and state the well-known sufficient condition for unique tensor decomposition in terms of the Kruskal rank of the factors.

**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). Suppose there exists an order \( k \) minimal HMM realization \( \theta^h = (d, k, Q, O) \) for the random process. Given the window size \( N = 2n + 1 \), let the matrices \( A, B, C \in \mathbb{R}^{d^n \times k} \) be functions of \( Q \) and \( O \) as in (24)–(26). The 3-rd order tensor \( M \) constructed as in (19)
can be uniquely decomposed into \( M = A \otimes B \otimes C \) up to column permutation and scaling, if
\[
\text{krank}(A) + \text{krank}(B) + \text{krank}(C) \geq 2k + 2.
\]
(27)

This is a direct application of the tensor decomposition uniqueness condition in [17], [20]. 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. In the following discussion, we will instead focus on the following conditions:

**Lemma 3** (Uniqueness of tensor decomposition). Given window size \( N \), if the matrices \( A, B, C \in \mathbb{R}^{d \times k} \) defined in (24)–(26) have full column rank \( k \), then \( M \) can be uniquely decomposed into column stochastic matrices \( A, B, C \) (up to common column permutation).

Next theorem shows that the above sufficient conditions are satisfied if the random process is generated by an HMM in general position and the window size \( N \) is sufficiently large.

**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 \( E \subset \Theta_{(d,k)}^h \) such that for all output processes generated by HMMs in the set \( \Theta_{(d,k)}^h \setminus E \), the information in \( P(N) \) is sufficient for finding the minimal HMM realization, for \( N = 2n + 1 \), with
\[
n > 2\lceil \log_d(k) \rceil.
\]
(28)

The proof is similar to that of Theorem 1 and is provided in the Appendix. We show that the matrices \( A \) and \( B \), as functions of the parameter \( Q \) and \( O \) of the minimal HMM realization, have full column rank \( k \). Moreover, when \( A \) and \( B \) have full column rank, Algorithm 2 finds the factors \( A, B, C \) uniquely, and then Theorem 4 recovers \( Q \) and \( O \) from the factors.

**D. Algorithms for minimal HMM realization**

When the window size is chosen to be large enough, we show how to find the minimal HMM realization from the tensor \( M \) in two steps.

1) **Recovering \( Q \) and \( O \) from \( A, B, C \):** As shown in (24)–(26), the matrices \( A, B \) and \( C \) are polynomial functions of the parameters \( Q \) and \( O \). The following theorem shows how to exploit the recursive structure of these functions to invert the polynomials efficiently.

**Theorem 4** (Recovering \( Q \) and \( O \) from \( A, B, C \)). Given the matrix \( C \), one can obtain the observation matrix by:
\[
O_{[i,:]} = C_{[i,:]} / (e^T C_{[i,:]}), \quad \forall i \in [k].
\]
(29)

Given the matrix \( A \in \mathbb{R}^{d^n \times k} \), we first scale each of the column similar to (29), so that each column is stochastic, and corresponds to the conditional probabilities \( P(y_1^n|x_0) \) as shown in (21). We marginalize the conditional distribution to get \( A^{(1)} = P(y_1|x_0) \in \mathbb{R}^{d \times k} \) and \( A^{(n-1)} = P(y_1^{n-1}|x_0) \in \mathbb{R}^{d^{n-1} \times k} \).

1) If \( A \) has full column rank \( k \):
\[
Q = \left( O \odot A^{(n-1)} \right)^\dagger A.
\]
(30)

2) If \( C \) has full column rank \( k \):
\[
Q = O^\dagger A^{(1)}.
\]
(31)

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

**Algorithm 2** Simultaneous diagonalization for 3-rd order tensor decomposition [18]

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

1) Randomly pick two unit norm vectors \( v_1, v_2 \in \mathbb{R}^d \). Project \( M \) along the 3-dimension to obtain two matrices:
\[
\tilde{M}_1 = M(I, I, v_1), \quad \tilde{M}_2 = M(I, I, v_2).
\]

2) Compute the eigen-decomposition of matrix \((\tilde{M}_1 \tilde{M}_2^{-1})\) and \((\tilde{M}_2 \tilde{M}_1^{-1})\), and let the columns of matrix \( A \) and \( B \) be the eigenvectors of \((\tilde{M}_1 \tilde{M}_2^{-1})\) and \((\tilde{M}_2 \tilde{M}_1^{-1})\), 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:
\[
\tilde{M}_1 \tilde{M}_2^{-1} = AA^{-1}, \quad \tilde{M}_2 \tilde{M}_1^{-1} = BB^{-1}.
\]

3) Let \( k \) be the number of non-zero eigenvalues.

4) Let \( \tilde{M}^{(3)} \in \mathbb{R}^{d^n \times d} \) be the 3-rd dimension matricization of \( M \). Set \( C \) to be:
\[
C = \tilde{M}^{(3)}((A \odot B)^\dagger)^T.
\]

b) **Tensor decomposition of \( M \):** Unlike singular value decomposition for matrices, no efficient algorithm is known for tensor decomposition in general [16]. Nevertheless, for cases where the factors \( A, B, C \) satisfy certain rank conditions, there are efficient and provable algorithms for tensor decomposition.

We have shown that for the output process of an HMM in general position, when the window size \( N > 4\lceil \log_d(k) \rceil + 1 \), the matrices \( A \) and \( B \) defined in (24)–(26), in terms of the parameter of the HMM that generates the random process, have full column rank \( k \). In this case, Algorithms 2 \[18\] can be applied to uniquely decompose \( M \) into the factors \( A, B, C \) up to common
column permutation, with running time polynomial in the dimension of the tensor \( M \in \mathbb{R}^{d \times d \times d} \), and thus polynomial in both \( d \) and \( k \) since \( n = 2 \lceil \log_2(k) \rceil \). Moreover, the first case in Theorem \( 4 \) can be applied to recover \( Q \) and \( O \) from the factors \( A, B \) and \( C \).

In practice, when \( k \) is not known a priori, an estimation of the upperbound of \( k \) is used to guide the selection of the window size. Algorithm \( 2 \) fails if the desired \( A \) and \( B \) are not of full rank, either because the window size is not large enough or because this is a degenerate case.

In particular, by definition of \( A \), if the transition matrix \( Q \) of the underlying minimal HMM realization does not have full rank, no matter how large the window size is, the matrix \( A \) is never of full rank. In this case of degeneracy, the tensor decomposition algorithm in Algorithm \( 2 \) may be applicable.

The basic idea of this 3-rd order tensor decomposition algorithm, first proposed in [11] and [14], is as follows: assume the tensor \( M \) of rank \( k \) can be decomposed into \( A \otimes B \otimes C \), and suppose that both the factor \( C \) and the Khatri-Rao product \( A \otimes B \) have full column rank \( k \), then there is a unique rank decomposition of the 3-rd dimension matricization of the tensor: \( \tilde{M}^{(3)} = FE^\top \), under the algebraic constraints that each column of the matrix \( E \) is a rank one Khatri-Rao product. In particular, the factor \( F \) and \( E \) correspond to the matrix \( C \) and \( A \otimes B \), respectively. For this algorithm, we fix \( n = 1 \), namely the window size \( N = 3 \). We also include the detailed steps of the algorithm here for completeness.

In order for Algorithm \( 3 \) to run correctly, additional conditions on the model parameter \( Q \) and \( O \) of the underlying minimal HMM realization need to be satisfied. For example, the observation matrix \( O \) needs to have full column rank \( k \) so that the matrix \( C \) has full column rank, which can be true only if \( d \geq k \). Note that the second case in Theorem \( 4 \) can be applied to recover \( Q \) and \( O \) once we obtain the factors \( A, B \) and \( C \).

Let \( \Theta_{(d,k,r)} \) denote the model class of HMMs with output alphabet \( d \) and order \( k \), for \( d \geq k \) and the transition matrix \( Q \) has rank \( r \). A natural question is, to which instances in this degenerate class of HMMs is Algorithm \( 3 \) applicable? The following theorem shows that if Algorithm \( 3 \) runs correctly for a random instance in this class, then the algorithm works for almost all HMMs in this class.

**Theorem 5 (Correctness of Algorithm \( 3 \)).** Given \( d, k \) and \( r \) and consider the set \( \Theta_{(d,k,r)} \). Let \( A, B, C \) be defined as in [24]–[26] for \( n = 1 \), and let \( M = A \otimes B \otimes C \). If Algorithm \( 3 \) returns “yes”, then there exists a measure zero set \( \mathcal{E} \subset \Theta_{(d,k,r)} \) such that Algorithm \( 3 \) returns the tensor decomposition \( M = A \otimes B \otimes C \) for all HMMs in the set \( \Theta_{(d,k,r)} \setminus \mathcal{E} \). Moreover, if the latter is true, Algorithm \( 3 \) returns “yes” with probability 1.

**Algorithm 3 FOOBI for 3-rd order tensor decomposition**

**Input:** \( M \in \mathbb{R}^{d \times d \times d} \)

**Output:** \( k, A, B, C \).

1. Let \( \tilde{M}^{(3)} \) be the 3-rd dimension matricization of \( M \). Compute its SVD \( \tilde{M}^{(3)} = V_H D_H U_H^\top \).
2. Set \( k \) to be the number of non-zero singular values.
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 \( \{ P^{(r,s)} \} \in \mathbb{R}^{d \times d \times d \times d} : r, s \in [k] \):
   \[ [P^{(r,s)}]_{i_1,i_2,j_1,j_2} = [E^{(r)}]_{i_1,j_1}[E^{(s)}]_{i_2,j_2} + [E^{(s)}]_{i_1,j_1}[E^{(r)}]_{i_2,j_2} - [E^{(r)}]_{i_1,j_2}[E^{(s)}]_{i_2,j_1} - [E^{(s)}]_{i_1,j_2}[E^{(r)}]_{i_2,j_1}. \]
4. Compute a basis \( \{ H^{(i)} \} : i \in [k] \) of the \( k \)-dimensional kernel of \( \{ P^{(r,s)} \} : r, s \in [k] \):
   \[ \sum_{r,s=1}^k H^{(i)}_{r,s} P^{(r,s)} = 0, \text{ s.t. } H^{(i)}_{r,s} = H^{(i)}_{s,r}, \forall r, s \in [k]. \]
5. Find the unique \( W \in \mathbb{R}^{k \times k} \) that simultaneously diagonalizes the basis:
   \[ H^{(i)} = WA^{(i)}W^\top, \forall i \in [k]. \]
6. Let \( C = F(W^{-1})^\top \) and \( A \otimes B = EW \). Compute the rank one decomposition of each column of \( A \otimes B \), with proper normalization such that \( A \) and \( B \) are column stochastic.

**Algorithm 4 Check Condition**

1. Randomly choose an HMM from \( \Theta_{(d,k,r)} \).
2. Construct matrices \( A, B, C \) with \( (Q, O) \) as defined in [24]–[26] for \( n = 1 \), namely \( A = OQ, B = OQ \), and \( C = O\text{Diag}(\pi) \).
3. Let \( M = A \otimes B \otimes C \). Run Algorithm \( 3 \) with the input \( M \).
4. Return “yes” if the algorithm returns \( A, B, C \) uniquely up to a common permutation, and “no” otherwise.

Note that both of the above two tensor decomposition algorithms involve mostly basic matrix operations of multiplications and inversions, as well as a common key step of simultaneous matrix diagonalization. For the computation and the first order perturbation analysis see [12].

Moreover, a detailed discussion on the sample complexity of Algorithm \( 2 \) can be found in Theorem 2.3 in [6], where it is shown that in order to
achieve $\epsilon$-accuracy in the output, the number of independent sample sequences needed is polynomial in \(1/d^n, 1/\sigma(E), 1/\epsilon, 1/\delta\), where $\delta$ is a measure of the incoherence of the factor $C$.

**V. Conclusion**

In this paper, we discussed two realization problems of Hidden Markov Models. 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, including the order the minimal realization.

**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] Anim 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, Ankur Moitra, and Aravindan Vijayaraghavan. Smoothed analysis of tensor decompositions. *arXiv preprint arXiv:1311.3651*, 2013.

[7] Aditya Bhaskara, Moses Charikar, and Aravindan Vijayaraghavan. Uniqueness of tensor decompositions with applications to polynomial identifiability. *arXiv preprint arXiv:1304.8087*, 2013.

[8] Anthony C. Bajgir and James Wright. Distributional and l’q norm inequalities for polynomials over convex bodies in $\mathbb{R}^n$. *Mathematical Research Letters*, 8(3):231–248, 2001.

[9] 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.

[10] Lieven De Lathauwer and Josephine Caiastraing. Tensor-based techniques for the blind separation of ds-cdma signals. *Signal Processing*, 87(2):322–336, 2007.

[11] Lieven De Lathauwer, Josephine Caiastraing, and J Cardoso. Fourth-order cumulant-based blind identification of underdetermined mixtures. *Signal Processing, IEEE Transactions on*, 55(6):2965–2973, 2007.

[12] Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. Computation of the canonical decomposition by means of a simultaneous generalized schur decomposition. *SIAM Journal on Matrix Analysis and Applications*, 26(2):295–327, 2004.

[13] 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.

[14] 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.

[15] 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.

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

[17] Joseph B Kruskal. Three-way arrays: rank and uniqueness of tri-linear decompositions, with application to arithmetic complexity and statistics. *Linear algebra and its applications*, 18(2):95–138, 1977.

[18] 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.

[19] 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.

[20] Nicholas D Sidiropoulos and Rasmus Bro. On the uniqueness of multilinear decomposition of n-way arrays. *Journal of chemometrics*, 14(3):229–239, 2000.

[21] Daniel A Spielman and Shang-Hua Teng. Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. *Journal of the ACM (JACM)*, 51(3):385–463, 2004.

[22] Sebastiaan A Terwijn. On the learnability of hidden markov models. In *Grammatical Inference: Algorithms and Applications*, pages 261–268. Springer, 2002.

[23] P Van Overschee and B De Moor. Subspace identification for linear systems: Theory, implementation, applications. 1996.

[24] Bart Vanluyten, Jan C Willems, and Bart De Moor. Equivalence of state representations for hidden markov models. *Systems & Control Letters*, 57(5):410–419, 2008.

[25] 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**

**Proofs**

(Proof of Lemma 1)

If both $E$ and $F$ have full column rank $k$, by Sylvester inequality the rank of the matrix $H(j)$ 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 2, there exists some full rank matrix $W \in \mathbb{R}^{k \times k}$ such that:

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

Therefore, Step 3 returns

$$\tilde{A}(j) = W^{-1}E A(j) F (F^T)^{1} W = W^{-1} A(j) W.$$ 

By the normalization constraint in Definition 1 we have

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

Moreover, since

$$U = \begin{bmatrix} u^T (A(1) \ldots A(1)) \\ u^T (A(1) \ldots A(2)) \\ \vdots \\ u^T (A(d) \ldots A(d)) \end{bmatrix},$$

$$W = \begin{bmatrix} \tilde{A}(1) \ldots \tilde{A}(1) \\ \tilde{A}(1) \ldots \tilde{A}(2) \\ \vdots \\ \tilde{A}(d) \ldots \tilde{A}(d) \end{bmatrix},$$

we have

$$u^T W = u^T W \sum_{j=1}^{d} \tilde{A}(j).$$
in Step 2 we obtain \( \tilde{w}^T = u^T W \) and similarly, we can argue that \( \tilde{v} = W^{-1} v \). Thus we conclude that the output \( \tilde{\theta} = (d, k, \tilde{u}, \tilde{v}, \tilde{A}^{(j)} : j \in [d]) \) is a valid minimal quasi-HMM realization of order \( k \).

\[ \tilde{\theta} \]

argue that \( \tilde{\theta} \) in Step 2 we obtain correctness of Algorithm 1 is that both \( E \) and \( F \) for \( \theta^o \) as in (11) and (12) and note that:

\[
E_{L, (v^0) : i} = [u^T (A^{(t_0)} \ldots A^{(t_i)})]_i = e^{y_n - x_n} \prod_{j=1}^{n-i} \mathbb{P}(x_j, y_0 = l_i | x_0 = i).
\]

and similarly,

\[
F_{L, (v^0) : i} = [A^{(t_0)} \ldots A^{(t_i) \pi}]_i = \mathbb{P}(y_{n-1}^0 = 1^1 x_0 = i).
\]

Lemma 1 shows that a sufficient condition for the correctness of Algorithm 1 is that both \( E \) and \( F \) have full column rank. In this proof, we show that when \( Q \) and \( O \) of the HMM \( \theta^h \in \Theta^h_{d,k} \) are in general position, this rank condition is satisfied if the window size \( N = 2n + 1 \) satisfies (16).

Since the minors of \( E \) and \( F \) are polynomials in the elements of \( Q \) and \( O \), it is enough to show that for some specific choice of \( Q \) and \( O \) in the parameter space, the matrices \( E \) and \( F \) achieve full column rank. Moreover, we can have \( Q \) and \( O \) to lie outside \( \Theta^h_{d,k} \) as long as it lies in the Zariski closure of \( \Theta^h_{d,k} \). Here we ignore the positivity and stochastic constraints.

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

\[
Q_i, i = 1, \quad 2 \leq i \leq k, \quad \text{and} \quad Q_{k+1, 1} = 1, \quad (32)
\]

Note that with this choice of \( Q \), \( \pi = \frac{1}{k} e \), 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:

\[
E_{[i, i]} = O_{[i, i]} \otimes \cdots \otimes O_{[i, i+n-1]}.
\]

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 \( O_{[j, j]} \), while for \( k < j \leq 2k \), it actually refer to the \((j - k)\)-th column of \( O \).

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

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

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

First suppose that, for any \( i, j \in [k] \) and \( i \neq j \),

\[
\mathbb{P} \left( \sum_{j \in [k], j \neq i} |X_{i,j}| < 1 \right) > 1 - \frac{1}{k^2}, \quad (33)
\]

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

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

Again apply union bound on \( i \), we have:

\[
\mathbb{P} \left( \forall i \in [k], \sum_{j \in [k], j \neq i} X_{i,j} < X_{i,i} \right) > \mathbb{P} \left( \forall i \in [k], \sum_{j \in [k], j \neq i} |X_{i,j}| < 1 \right) > 1 - \frac{1}{k} = 0.
\]

By Gershgorin’s theorem, with non-zero probability, the matrix \( X = E^\top E \) is of full rank \( k \), thus there must exist at least one \( O \) such that \( E \) achieves full column rank.

Next, we verify the statement in (33). Equivalently, we want to show that for \( i \neq j \):

\[
1 - \frac{1}{k^2} < \mathbb{P} \left( \prod_{m=0}^{n-1} O_{[i, i+m]} O_{[j, j+m]} \right) < \frac{1}{k^2}
\]

\[
= \mathbb{P} \left( \sum_{m=0}^{n-1} \log \left( O_{[i, i+m]} O_{[j, j+m]} \right) < - \log(k) \right)
\]

\[
= \mathbb{P} \left( \sum_{m=0}^{n-1} \log \left( \frac{1}{|v_m|} \right) > \log(k) \right)
\]

where \( v_m \) are independent random variables with the same distribution as the projection of a uniformly distributed unit-norm vector in \( \mathbb{R}^d \) onto the first dimension. The last equality is due to the fact that the columns of
Define the indicator random variable $s_m$ for $m \in [n]$:

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

where $c$ is a constant and $c = 2$. Then 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

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

or equivalently,

$$\frac{1}{k^2} > \mathbb{P} \left( \sum_{m=1}^{n} s_m > n - c \frac{\log (k)}{\log (d)} \right) \quad (34)$$

$$= \mathbb{P} \left( \sum_{m=1}^{n} s_m > (a-c) \frac{\log (k)}{\log (d)} \right) \quad (35)$$

where we assume that $n = a \log_d (k)$ for $a > c$.

Assume that $d \geq 3$, apply Johnson Lindenstrauss lemma of high dimensional sphere projection (Lemma 9), setting $u_1$ to be $v_m$, and $t$ to be $1/d^c$, we have:

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

Apply the multiplicative Chernoff bound (Lemma 8), by setting $X_m = s_m$ for $m = 1, \ldots, n$, and set

$$ \delta n \mu = (a-c) \frac{\log (k)}{\log (d)} \quad (a-c) \frac{\log (k)}{\log (d)} \quad (36)$$

then we have

$$ \mathbb{P} \left( \sum_{m=1}^{n} s_m > (a-c) \frac{\log (k)}{\log (d)} \right) < \left( \frac{e \mu}{a-c} \right)^{(a-c) \frac{\log (k)}{\log (d)}} \quad (37)$$

We want to show that the RHS is less than $1/k^2$. Taking log with base $d$, this is equivalent to:

$$ (a-c) \frac{\log (k)}{\log (d)} \left( \log (d) \mu + \log_d \left( \frac{e \mu}{a-c} \right) - 2 \frac{\log (k)}{\log (d)} \right) < 0.$$

Note that since $\mu < 4 \exp(-\frac{d-2}{2k^2})$ and $c = 2$, we have $\log_d(d) < \log_d(k) - \frac{1}{2} \frac{d^2}{d^2} \log (d)$. Then the above inequality holds if

$$(a-c) \left( - \frac{1}{2} \frac{d^2}{d^2} \log (d) + \log_d \left( \frac{4e \mu}{a-c} \right) \right) - 2 < 0.$$

Note that the LHS is decreasing in $d$, it is straightforward to verify the inequality by setting $a = c+\epsilon$ for any $\epsilon > 0$. Now we can conclude that $\mathbb{P}$ holds.

(Proof to Theorem 2)

Recall that the output of Algorithm 1 is given by:

$$\hat{A}^{(j)} = \hat{D}^{-1/2} \hat{U}_H^{(j)} \hat{H}^{(j)} \hat{V}_H \hat{D}^{1/2},$$

$$\hat{u} = \hat{D}^{-1/2} \hat{U}_H \mathbf{e}, \quad \hat{v} = \hat{D}^{-1/2} \hat{V}_H \mathbf{e},$$

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 $A^{(j)}$, $\hat{u}$ and $\hat{u}$, $\hat{v}$ and $\hat{v}$, 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 $j = 0, 1, \ldots, d$.

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

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

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

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

$$||\hat{U}_H - U_H||_F \leq \sqrt{2}\|E_0||_F, \quad ||\hat{V}_H - V_H||_F \leq \sqrt{2}\|E_0||_F.$$

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

$$\left( \frac{1}{\sqrt{\sigma_i}} - \frac{1}{\sqrt{\sigma_i + \Delta_i}} \right)^2 = \frac{1}{\sigma_i + \Delta_i} \left( \sqrt{1 + \Delta_i/\sigma_i} - 1 \right)^2$$

$$\leq \frac{2}{\sigma_i} (\Delta_i/\sigma_i + 2 - \sqrt{1 + \Delta_i/\sigma_i})$$

$$\leq \frac{2}{\sigma_i} (3||\Delta_i||/\sigma_i)$$

$$\leq \frac{6}{\sigma_k} ||\Delta_i||,$$

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

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

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
\[
\|\tilde{A}^{(j)} - A^{(j)}\| \\
\leq 2^{4} \left( \frac{2 \sqrt{k} - 1/2 \|E_0\|_F}{\sigma_k^{1.5}} + \frac{2 \sqrt{2} \|E_0\|_F}{\sigma_k^{1.2}} + \frac{\|E_j\|}{\sigma_k} \right) \\
\leq 2^{4} \left( \frac{2 \sqrt{k} \delta \sigma_k^{0.5} d^{-0.5}}{\sigma_k^{1.8}} + \frac{2 \sqrt{2} k \delta \sigma_k^{0.5} d^{-0.5}}{\sigma_k^{2}} + \frac{kd \delta \sigma_k^{0.5}}{\sigma_k} \right) \\
\leq 144kd \delta \sigma_k^{0.5} d^{-0.5},
\]
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 \( \|\tilde{u} - \tilde{u}\| \) and \( \|\tilde{v} - \tilde{v}\| \) by:
\[
\|\tilde{u} - \tilde{u}\| \leq \|\tilde{D}^{1/2} \tilde{U}_{H}^{\top} - D^{-1/2} U_{H}^{\top}\| \sqrt{d^m} \leq 4 \frac{d^{1.5} d^{-0.5}}{\sigma_k^{1.5}} \delta d^{-0.5},
\]
In summary, if we want to achieve \( \epsilon \) accuracy in the output, we need \( \delta \) to be no larger than \( \frac{c^2 \sigma_k^4}{(144kd^2 d^2)} \).
Set the failure probability to be \( \eta = 2k^4d^3 \epsilon^2 \), then number of sample sequences needed to estimate the empirical probabilities is given by:
\[
T = 2 \frac{144k^4 d^4 \epsilon}{c^2 \sigma_k^4} \log \left( \frac{2k^4 \delta^2 d^2}{\eta} \right).
\]

**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 (25), 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: \( \vec{e}^{\top} O_{[i,j]} = 1, \forall i \in [k] \). The normalization in (29) recovers \( O \) from \( C \).

Recall that
\[
A = A^{(n)} = \left( O \circ A^{(n-1)} \right) Q,
\]
Since the matrix \( A \) has full column rank \( k \), the matrices \( Q \in \mathbb{R}^{k \times k} \) and \( O \circ A^{(n-1)} \in \mathbb{R}^{d^2 \times k} \) both have full column rank \( k \), as well as the pseudo-inverse of \( (O \circ A) \), therefore \( Q = (O \circ A^{(n-1)})^{-1} A \).

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

**Proof of Theorem 5**

Denote the minimal order HMM realization by \( \theta^h = (d, k, Q, O) \), and since \( n = 1 \), the matrices are given by:
\[
A = OQ, \quad B = O\hat{Q}, \quad C = O\text{Diag}(\pi).
\]
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 \( Q \in \mathbb{R}^{d^2 \times d^2} \) 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 \( G^\perp \in \mathbb{R}^{d^4 \times (d^4-d^2(d-1)^2)/4} \) such that its columns are orthogonal to the columns of \( G \).

According to [9], [13], there are two deterministic conditions for Algorithm 3 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 \( T \in \mathbb{R}^{d^2 \times (m+(k-1)k/2)} \) to be:
\[
T = \left[ G_{[i,j]}^\perp : 1 \leq i \leq d^2 - \frac{d^2(d-1)^2}{4}, \right.
A_{[i,k_1]} \odot A_{[i,k_2]} \odot B_{[i,k_1]} \odot B_{[i,k_2]} : 1 \leq k_1 < k_2 \leq k \].
\]
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 \( Q \):
\[
Q = \{ Q \in \mathbb{R}^{k \times k} : Q = UV^\top, U, V \in \mathbb{R}^{k \times r}, e^T Q = e^T \}
\]
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 \( \Theta^h_{(d,k,r)} \), it is enough to construct an instance in the model class that satisfies the two conditions (by the random check in Algorithm 4). Moreover, if it is true, then with probability one, the two conditions are satisfied for a randomly chosen instance in the model class.

**Appendix B**

**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 \( \hat{A} = A + E \) 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

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

**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 \( \hat{A} = U \Sigma V^T \) 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:

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

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 \( \hat{A}_1, \ldots, \hat{A}_k \) and assume that \( \| \hat{A}_i - A_i \| \leq \| A_i \| \), then we have:

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

(Concentration bounds)

**Lemma 7** (Hoeffding’s inequality). Let \( X_1, \ldots, X_n \) 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 \( \hat{X} = (X_1 + \cdots + X_n)/n \). We have

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

**Lemma 8** (Multiplicative Chernoff bound). Suppose \( X_1, \ldots, X_n \) 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 \( u \in \mathbb{R}^d \) be uniformly distributed on the surface of the \( d \)-dimensional unit sphere, i.e. uniform distribution in the set: \( \{ \sum_{i=1}^{d} u_i^2 = 1 \} \). Denote its projection onto the first dimension to be \( |u_1| \). We have:

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

**Lemma 10** (Gershgorin’s theorem). Given a symmetric matrix \( X \in \mathbb{R}^{k \times k} \), a lower bound on the smallest eigenvalue is given by:

\[
\sigma_{\min}(X) \geq \min_{i \in [k]} \left\{ X_{i,i} - \sum_{j \in [k], j \neq i} |X_{i,j}| \right\}.
\]