MARKOV APPROXIMATIONS: THE CHARACTERIZATION OF UNDERMODELING ERRORS

by

Lei Lei

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computer Science
Brigham Young University
August 2006

BRIGHAM YOUNG UNIVERSITY

GRADUATE COMMITTEE APPROVAL

of a thesis submitted by

Lei Lei

This thesis has been read by each member of the following graduate committee and by majority vote has been found to be satisfactory.

Date Sean C. Warnick, Chair

Date Kevin D. Seppi

Date Irene L. Geary

BRIGHAM YOUNG UNIVERSITY

As chair of the candidate's graduate committee, I have read the thesis of Lei Lei in its final form and have found that (1) its format, citations, and bibliographical style are consistent and acceptable and fulfill university and department style requirements; (2) its illustrative materials including figures, tables, and charts are in place; and (3) the final manuscript is satisfactory to the graduate committee and is ready for submission to the university library.

Date

Sean C. Warnick
Chair, Graduate Committee

Accepted for the Department

Parris K. Egbert
Graduate Coordinator

Accepted for the College

Thomas W. Sederberg, Associate Dean
College of Physical and Mathematical Sciences

ABSTRACT

MARKOV APPROXIMATIONS: THE CHARACTERIZATION OF UNDERMODELING ERRORS

Lei Lei

Department of Computer Science

Master of Science

This thesis is concerned with characterizing the quality of Hidden Markov modeling when learning from limited data. It introduces a new perspective on different sources of errors to describe the impact of undermodeling. Our view is that modeling errors can be decomposed into two primary sources of errors: the approximation error and the estimation error. This thesis takes a first step towards exploring the approximation error of low order HMMs that best approximate the true system of a HMM. We introduce the notion minimality and show that best approximations of the true system with complexity greater or equal to the order of a minimal system are actually equivalent realizations. Understanding this further allows us to explore integer lumping and to present a new way named weighted lumping to find realizations. We also show that best approximations of order strictly less than that of a minimal realization are truly approximations; they are incapable of mimicking the true system exactly.

Our work then proves that the resulting approximation error is non-decreasing as the model order decreases, verifying the intuitive idea that increasingly simplified models are less and less descriptive of the true system.

ACKNOWLEDGMENTS

I really appreciate all the great efforts my advisor Dr. Sean Warnick went into in funding my education and guiding me throughout my research patiently and energetically. I would like to thank Dr. Kevin Seppi and Dr. Irene Geary for their time and efforts put on this thesis. I thank all the members of the IdEA lab for generously offering me suggestions and help.

I also would like to thank my parents, my boyfriend and my Chinese friends for their love, support and encouragement throughout my education.

TABLE OF CONTENTS

Section Page

1 Overview 1

1.1 Background 4

1.2 Previous Work 10

1.3 Thesis Statement 13

1.4 Contribution 13

2 Realizations of Non-minimal Systems 15

2.1 Low-order Realizations of Non-minimal Systems 17

2.1.1 Integer Lumping 22

2.1.2 Problems with Lumping 32

2.1.3 Weighted Lumping 37

2.1.4 Realizations with Weighted Lumping 43

3 Approximations of Minimal Systems 51

3.1 Lower-order Approximations of Minimal Systems 51

4 Conclusions and Future Work 57

4.1 Conclusions 57

4.2 Future Work 58

5 Appendix 61

5.1 Transform a n^{th} order difference equation to n first-order equations . . . 61

5.2 Derivation of the second condition in general lumping 62

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

1.1 **Relationship between modeling errors of different systems** 3

1.2 **A HMM can be regarded as an automaton** 7

2.1 **An automaton of the ordinary lumping example** 26

2.2 **A matrix representation of the ordinary lumping example** 28

2.3 **A matrix representation of the exact lumping example** 29

2.4 **An automaton of the exact lumping example** 30

2.5 **Output distributions of two HMMs** 34

2.6 **Systems of different orders** 35

2.7 **Relationship of different lower order system sets** 40

3.1 **Approximation errors of different order systems** 55

4.1 **The whole picture of three types of modeling errors** 59

Chapter 1

Overview

When learning from limited data, we generally would like to characterize the quality of our efforts. Learning algorithms use a finite data record to select a mathematical description, or model, of the ”true system” generating the data from a class of candidate models. Typically the resulting model is only an approximation of the true system, and it is often essential to understand the nature of errors in this model before using it as the basis for future decisions, or as the basis of understanding of the underlying phenomenon.

Characterizing modeling errors in general is quite difficult. This is because the impact of different sources of errors are not well understood. This work introduces a perspective on the topic that exposes it to a rigorous investigation. Our view is that modeling errors can be decomposed into two primary sources of errors: the approximation error and the estimation error. First, an *approximation error* is a measure of the distance between the true system and the set of simplified candidate models. It is often unavoidable because the class of candidate models that our learning algorithms explore are simple compared to the true system, which is assumed to be complex. It is also noted that this error is independent of the learning algorithm or the amount of data available. Second, an *estimation error* is a measure of the distance between the estimated model and the set of candidate models. It is also often unavoidable since the learning algorithm only has access to a finite data record, assumed to be insufficient compared to the complexity of the underlying phenomenon. Unlike the approximation error, this error not only depends strongly on the learning algorithm and the amount of data available, but may depend on the class of candidate models i.e. on the approximation error as well. In this sense the notion of approximation error is fundamental to understanding the estimation error and, ultimately, the complete modeling errors for any process claiming to learn from limited data.

This thesis takes a first step towards understanding these ideas by characterizing the nature of approximation error for hidden Markov models. A *hidden Markov model* (HMM) is a discrete-time finite-state Markov chain observed through a discrete-time memoryless channel. The channel can be thought of as a (possibly probabilistic) mapping of the state into the output; the resulting output stochastic process dose not need to be Markovian. Such processes are extremely general and have been used in a variety of applications including automatic speech recognition [1], language modeling [2], communications and information theory [3], econometrics [4] and biological signal processing [5]. Our interest in limiting the scope of our study to such processes is that these processes have a clear notion of complexity, namely the model *order*, or number of hidden states in the HMM state vector.

With this clear definition of model complexity, we can then rigorously investigate the nature of approximation error in this context. Given a high-order HMM, i.e. the "true system", we are interested in characterizing the approximation error of low-order HMMs that best approximate this true system (see Figure 1.1). This clearly demands a notion of comparison between models of different orders. We introduce such a notion of comparison and then use this notion to define the concept of a *minimal realization* of a given HMM, that is, the simplest HMM that is equivalent to the true system (i.e. zero error as measured by our notion of comparison between systems).

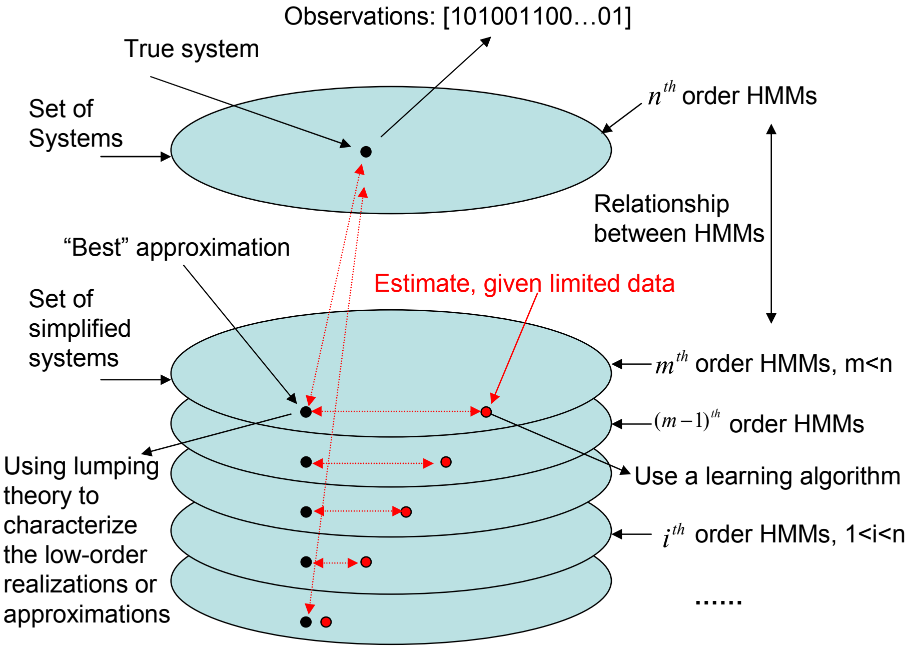

Observations: [101001100…01]
True system
Set of Systems
n^{th} order HMMs
Relationship between HMMs
"Best" approximation
Estimate, given limited data
Set of simplified systems
m^{th} order HMMs, m<n
$(m-1)^{th}$ order HMMs
Using lumping theory to characterize the low-order realizations or approximations
Use a learning algorithm
i^{th} order HMMs, 1<i<n
……

Figure 1.1: **Relationship between modeling errors of different systems**

We show that best approximations of the true system with complexity greater or equal to the order of a minimal realization are actually not approximations at all, but equivalent realizations. Understanding this fact allows us to interpret certain clustering results on lumpable Markov processes found in the literature and to develop new equivalence results for such systems as alternate realizations of the same process. On the other hand, we find that best approximations of order strictly less than that of a minimal realization are truly approximations; they are incapable of mimicking the true system exactly. Our work then proves that the resulting approximation error is non-decreasing as the model order decreases, verifying the intuitive idea that increasingly simplified models are less and less descriptive of the true system.

The thesis is organized as follows. The remainder of this chapter provides background information about Markov models and HMMs, realization theory including positive linear systems and lumping theory as well as current work on approximations.

Chapter 2 explains the relationship between high-order realizations of non-minimal systems. To find out the realizations of non-minimal systems, we first explain in details the lumping theory from previous work and then develop our own method named weighted lumping to extend the applicability of lumping theory. Chapter 3 demonstrates the relationship between low-order approximations of minimal systems and provides one of our critical proofs that the distance of low-order HMMs to a minimal HMM system is monotonically nondecreasing as the order decreases. Chapters 4 gives our conclusions and future work.

1.1 Background

Markov processes and hidden Markov models (HMMs) are widely used in numerous domains, including bioinformatics [5], web traffic control [6], speech recognition [7] and ecological species population prediction [8]. A Markov process is a random process in which the current state in the process only depends on the previous state [9]. In the case where the process can assume only a finite or countable set of sets, it is a *Markov chain* [10]. Although there are various types of Markov models, this thesis will primarily focus on discrete-time finite state space Markov models.

Denote *path* $Q = q_{(1)}, q_{(2)}, ...$ as a discrete-time sequence of random variables and $q_{(k)}$ as the k^{th} *state* in the path with a value in a finite *state space* $S_q = \{1, 2, \ldots, n\}$.

Definition 1 A sequence $q_{(1)}, q_{(2)}, ...$on S_q is a Markov chain if it preserves the *Markovian* property, that is, if for all $t > 0$, $i, j \in S_q$,

$Pr(q_{(k)} = j | q_{(k-1)} = i_{k-1}, q_{(k-2)} = i_{k-2}, \ldots, q_{(1)} = i_1) = Pr(q_{(k)} = j | q_{(k-1)} = i_{k-1})$

A Markov chain is *homogeneous* if the *one step transition probability* $Pr(q_{(k)} = j | q_{(k-1)} = i)$ is independent of time k. Denote the transition probability from j to i by $a_{ij} \triangleq Pr(q_{(k)} = i | q_{(k-1)} = j)$. The matrix $A \triangleq \{a_{ij}, i, j = 1, 2, \ldots, n\}$ is called *transition matrix* [11].

A transition matrix A will always be a stochastic matrix, meaning that either every column of it sums to one (i.e. it is column stochastic), or every row of it sums to one (i.e. it is row stochastic), and each of its entries are between zero and one.

Column stochastic and row stochastic matrices are equivalent in representing the Markovian property and can be changed to the other through linear transformations. Although some work in the literature uses row stochastic matrices to represent the dynamics of a Markov chain, this thesis represents Markov models using the column stochastic convention.

The *Hidden Markov Model (HMM)* is a generalization of Markov chains that captures the dynamics of observational sequences and "hidden" sequences. The hidden sequence is a Markov chain while the observational sequence itself is not necessarily Markovian . HMMs were introduced in 1966 by Baum and Petrie [12] who studied statistical properties of stationary ergodic finite-state finite-alphabet HMMs and proved consistency and asymptotic convergence of Maximum Likelihood estimation [9]. Petrie also provided sufficient conditions for identifiability of a HMM in [13].

In a HMM, there are two fundamental parameter spaces: a group of hidden states and a collection of output symbols. Each state generates a symbol from a specified distribution over all possible symbols. Describe the observational sequence O over a finite alphabet $S_o = \{0, 1, 2, \ldots, m\}$ as $O = o_{(1)}, \ldots, o_{(T)}$ where $o_{(1)}, \ldots, o_{(T)} \in S_o$ and T is the sequence length.

To distinguish symbols from states, *emission* probability c_{ij} is introduced and defined as the probability that a symbol $i \in S_o$ is observed when in state j:

$$c_{ij} = Pr(o_{(k)} = i | q_{(k)} = j) \tag{1.1}$$

The matrix $C \triangleq \{c_{ij}, j = 1, 2, \ldots, n; i = 1, 2, \ldots, m\}$ is called *emission matrix* [14], where the j^{th} column, c_j, is the output emission probability mass function over S_o given the state current $q = j$.

The hidden sequence Q is a Markov chain; however, it is well known that the stochastic observational sequence O may not be a Markov process [15], but simply a function of the hidden states.

Definition 2 A path Q and an observational sequence O will form a *hidden Markov model* if they satisfy the following relations:

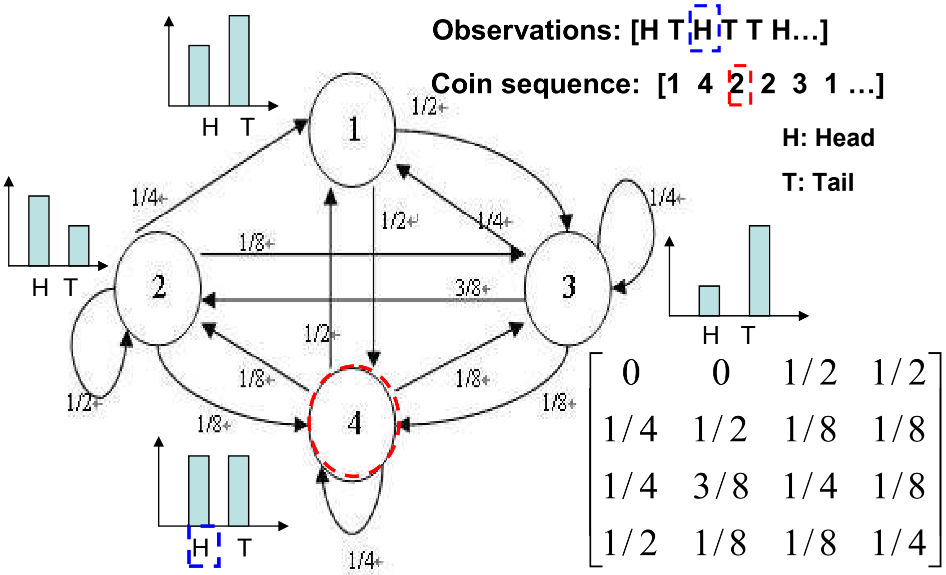
$$Pr(q_{(k)}|q_{(1)},\ldots,q_{(k-1)},o_{(1)},\ldots,o_{(k-1)}) = Pr(q_{(k)}|q_{(k-1)})$$

$$Pr(o_{(k)}|q_{(1)},\ldots,q_{(k-1)},q_{(k)},q_{(k+1)},\ldots,q_{(T)},o_{(1)},\ldots,o_{(k-1)},o_{(k+1)},\ldots,o_{(T)}) = Pr(o_{(k)}|q_{(k)})$$

The process of generating a sequence from a HMM is as follows: Given an initial state $q_{(0)}$, a state $q_{(1)}$ is then selected with the probabilities a_{10}. In that state, an observation is emitted according to the distribution $c_{q_{(1)}}$. Then a new state, $q_{(2)}$, is chosen according to the transition probabilities in A and a new observation is emitted according to $C_{q_{(2)}}$, and so on [16]. In this way, a random sequence of observations is produced.

A HMM can also be regarded as an automaton (See Figure 1.2). The states in the state vector can be regarded as nodes in the automaton. Each node can have incoming and outing arrows. A number a_{ij} on an outgoing arrow represents a transition probability from node j to i. If $a_{ij} = 0$, it means that there is no transition from j to i. In this situation, we do not put a_{ij} on this outgoing arrow. At each node, an observation will be emitted according to a distribution associated with that node. This represents the emission transition probability.

Although a HMM defines the behavior of two stochastic process, Q and O, it is convenient to analyze the dynamics of these processes in forms of the evolving probabilities, not in forms of the actual random variables q and o. Denote the set $\mathbb{R}_+ = [0, +\infty)$ as the positive real numbers, $\mathbb{R}^{n\times m}$ as the set of matrices of size n-by-m, and $\mathbb{R}^n$ as the set of vectors of size n-by-1. We then define the probability mass function from which the state $q_{(k)}$ is sampled as $x(k) \in \mathbb{R}$. That is, $x_i(k)$ represents



$$\begin{bmatrix} 0 & 0 & 1/2 & 1/2 \\ 1/4 & 1/2 & 1/8 & 1/8 \\ 1/4 & 3/8 & 1/4 & 1/8 \\ 1/2 & 1/8 & 1/8 & 1/4 \end{bmatrix}$$

This is an coin flip example of an HMM. There are four numbered coins in the pocket. One is tossed each time. An underlying state sequence consists of a series of coin labels. The numbers on the arrows are the transition probabilities from one state to the other. What you can observe is a sequence of heads and tails. The histograms show individual distributions of heads and tails at each state.

Figure 1.2: **A HMM can be regarded as an automaton**

the probability that $q_{(k)} = i$, $\{i = 1, \ldots, n\}$. The state transition matrix A then completely characterizes the dynamics of $x(k)$ as

$$x(k+1) = Ax(k) \qquad x(0) = x_o$$

where $x(0)$ or x_o is the initial state distribution. Likewise, at time k, let $y(k) \in \mathbb{R}^m$ be the distribution from which $o_{(k)}$ is sampled. Then we have

$$y(k) = Cx(k)$$

where C is the emission matrix. The dynamics of a HMM are thus completely characterized by a positive linear dynamic system

$$\begin{aligned} x(k+1) &= Ax(k) \qquad x(0) = x_o \\ y(k) &= Cx(k) \end{aligned}$$

in which $A \in \mathbb{R}_+^{n \times n}$, $C \in \mathbb{R}_+^{r \times n}$, $x(k) = \mathbb{R}_+^n$ and $y(k) = \mathbb{R}_+^r$.

Note that this is a discrete-time noise free zero-input HMM. A comprehensive study on the dynamic system setup of HMMs is also provided by Elliott in [17].

The number of states in $x(k)$ of an HMM is called the *order*. Denote $\mathbf{1}$ as the vector $[1\ 1 \ldots 1]^T$ with corresponding dimensions, we have

$$x(k)^T\mathbf{1} = \mathbf{1} \qquad y(k)^T\mathbf{1} = \mathbf{1}$$

$$A^T\mathbf{1} = \mathbf{1} \qquad C^T\mathbf{1} = \mathbf{1}$$

If the initial condition $x(0)$ and transition and emission matrices A and C are known, $x(k)$ and $y(k)$ at time step k can be calculated as:

$$x(k) = Ax(k-1) = A \cdot Ax(k-2) = \ldots = A^k x(0)$$

$$y(k) = Cx(k) = CA^k x(0)$$

Since a pair of transition and emission matrices describe all the dynamics of a system, we use (A, C) to represent a HMM in the thesis. If a specific path and an observational sequence are noted, then the corresponding initial condition $x(0)$ would also be relevant.

Note, in the community of difference equations, the ”order” of a difference equation could also refer to the difference between the highest and lowest indices in the equation [18]. For example, $x(k+3) = Ax(k)$ is a third-order system. However, one n^{th} order difference equation can always be turned into n first-order difference equations [19] (see proof in the Appendix 5.1). Thus, they are equivalent. Without loss of generality, we confine our attention to the first-order difference equations and use the word ”order” in this thesis to mean the number of states in the hidden state vector of a first-order state difference equation. For systems of different orders, we use *true system* to refer to the n^{th} order HMM that generates data, and *reduced* or *low-order system* to refer to m^{th} order HMMs, where $m < n$. With the clarification, we can now use ”order” to examine the complexity of an HMM.

Order is a good measure of complexity for HMMs because it governs how many parameters are needed to completely characterize the dynamics of a HMM. The number of parameters in a n^{th} order HMM is defined as the total number of parameters in the transition and emission matrices. We do not include the initial condition of the hidden state vector in our parameters unless a specific path or sequence is necessary. In the thesis, the observation space is chosen to be binary, i.e $S_o = \{0, 1\}$, although all the results generalize. With the binary observation space, the number of parameters that needs to be estimated in a n^{th} order HMM is $n \times n + 2 \times n = n^2 + 2n$. It is a function of the order and indicates that the order of a system restricts the number of parameters that can be estimated. This restriction causes a problem in practice when people want to use a learning algorithm to estimate the parameters but have no information on the order of the true system. Even with a well-performing learning algorithm, it is still very likely that people underestimate the true system because they select an order less than that of the true system to build their model.

A practical example of this problem can be found from applications of Markov processes in forestry management for landscape diversity under uncertainty. An appropriate forestry management plan has to consider both ecological concerns and economic expectations. Modeling the ecological effects of a management strategy on the landscape is primarily restricted by the need to keep the number of system states such as species composition and possible catastrophic disturbances small enough for numerical solutions. This may result in an oversimplification of the biological system [20]. However, it is necessary to simplify the definitions of states in order to carry out a practical silvicultural prescription. Thus, a choice of an appropriate number of states or the order of the system becomes vital before we can focus on details of other parameters in the model. It is also important to evaluate how well simplified models can perform compared to the true model in predicting different silvicultural pathways and consequences of various treatments. It is desired that a low-order less computational intensive model be built to approximate the true model as close as possible.

1.2 Previous Work

A review of order estimation approaches can be found in [9]. Finesso developed an information-theoretic approach to estimate the order of a finite-alphabet HMM in [21]. Kieffer proposed a code-based approach to estimate the order of time invariant ergodic finite-state HMMs and proved strong consistency of the algorithm [22]. Ziv and Merhav used a Neyman-Pearson type criterion to minimize the probability when the estimated order is less than the true order in [23]. However, they all estimated the order simply based on their learning algorithms especially on the Maximum Likelihood algorithm. This is different from our work. We not only examine the connection between true system and estimations, but also the relationship between approximations and estimations that does not depend on the learning algorithm. In addition, we attempt to disclose the uniqueness of an approximate order by considering the tradeoff between uncertainty and complexity, which is explained by two different modeling errors.

Besides exploring the tradeoff, we study the characteristics of different true systems and identify their possible bounds. This is a minimality problem. Since equivalent representations, i.e. systems of order equal or higher than a minimal system always exist (we will explain this more in the later chapters), the first step towards solving a minimality problem is to characterize the minimal order of one system whose behavior is consistent with the observational information. It is within the scope of realization theory.

The realization problem for HMMs is: under what circumstances two finite-state HMMs can generate the same output distribution? In other words, given a HMM and its initial condition, find a way to construct another HMM (the transition, emission matrices and possibly the initial condition if required) such that this constructed HMM will be able to produce an identical observational distribution sequence as the true system. Particularly, it is interesting to characterize the minimal requirement for two systems to be eligible for equivalence, the notion of which we will explain later. In 1957, Blackwell and Koopmans first discussed the realization problem in [24]. Picci studied the stochastic realization problem of finite-valued processes in [25]. Hof and Schuppen provided sufficient conditions for the existence of a positive realization by means of polyhedral cones [26]. They considered non-zero input dynamic systems and their sufficient condition is stronger than the reachability/observability condition. However, the Hankel matrix in their sufficient condition becomes a zero matrix when the input is zero and thus does not apply to the HMMs we discuss in this thesis. They also mentioned that classification of all minimal positive realizations as well as factorization of positive matrices remained as the two unsolved issues in characterizing the minimality of systems [26]. A complete solution to the realization problem has not been found. Blackwell and Koopmans provided a solution to the realization problem under special conditions [24]. Gilbert introduced LDP (Linearly Dependent Processes) to extend the generality of the solution, but it is still a partial solution [27]. Ito, Amari and Kobayashi used algebraic systems, equivalent relations and quotient algebraic systems established by Kalman, Falb and Arbib [28] to solve the minimality problem in [15]. They gave a necessary and sufficient condition to identify the equivalence of two HMMs, but their approach merely worked with one single fixed initial condition and could not be generalized to various initial conditions. Partial solutions can also be found in [29] and [30]. It is the nonlinearity caused by the positivity constraint of probability and the non-observability of states that make the realization problem demanding to solve [15].

To consider a minimality problem, research usually starts with realizations of non-minimal systems. Lumping is an approach to obtain a realization of a non-minimal system. It uses either an integer or real aggregation operator to map the true system to the low-order system. Actually lumping is a special kind of aggregation in terms of preserving the Markovian property in the low-order systems. The Simon-Ando theory in [31] first discussed the aggregation of variables in the nearly decomposable dynamic systems and provided an approach to represent a system as a superposition that separated the short-run and long-run dynamics. This laid the theoretical foundation for most aggregation techniques [32]. The approach developed in [33] simplified a Markov chain by removing irrelevant states calculated through a reward function [34] when the function scores were below the predefined threshold. The concept of lumpability has been addressed in [35] to discuss finite Markov chains. Model reduction with irreducible Markov chains can also be found in [36]. Only recently has lumping been applied to the area of HMMs [37]. Concepts of lumpability of HMMs were generalized in [37] and necessary and sufficient conditions were suggested for HMMs to be lumpable. Three fundamental lumping algorithms were presented in [38] for discrete Markov chains based on their time variance property and the direction of transitions. Ito et al. provided sufficient and necessary conditions for the identifiability problem of Markov chains though they did not explicitly use the notion of lumping [15]. We show in the next chapter that all types of lumpings are simply methods to generate equivalent realizations of non-minimal systems.

While the realization problem examines the situations when two HMMs can generate identical output distributions, an approximation problem aims at characterizing a minimal difference between the output distributions of two HMMs of different orders. It is similar to the notation of a realization problem except for that it does not guarantee a zero difference between the true system and the approximated system. Until now not much has been done in finding a low-order system which miniminzes modeling errors to the true system. Kotsalis and Dahleh discussed the approximation problem between a reduced system and the original system and provided a metric to measure the asymptotic distance of two irreducible Markov chains [36]. Beheshti also presented Minimum Description Complexity (MDC) as a new order estimation method to identify a best representative model of the true LTI (Linear

Time-Invariant) system among a set of candidate models [39]. She examined the role of undermodeling no matter the true system is included in the competing model sets or not. This is similar to what we consider in this thesis where we mostly investigate the situation when the order of a candidate model is less than the true system.

1.3 Thesis Statement

In this thesis, we propose a decomposition of Hidden Markov modeling errors into two pieces: approximation errors which are independent of data and describe the difference between a true system and a system of order less than the true system; and estimation errors which are caused by learning algorithms given limited data. We further decompose the approximation errors into realizations and best approximations, which characterize the situations of zero and non-zero distances between systems respectively. We seek to characterize minimality and use lumping theory to find realizations of non-minimal low order systems. We also seek to characterize the relationship between best approximations and the true system as the order of simplified systems decreases.

1.4 Contribution

In this thesis, we introduce decomposition of modeling error in two pieces: approximation errors and estimation errors. We first define what we mean by of distance, dynamic consistency, realization and best approximation. We then characterize minimality and show the proof of a known result that a system is minimal if it is observable. We show that best approximations of a true system with complexity greater or equal to the order of a minimal realization are actually equivalent realizations of the true system. We prove this by embedding a true system into high-order equivalent systems to construct realizations. To find low-order realizations of non-minimal HMMs, we present three types of integer lumping from previous work and discuss their weaknesses. We further develop an approach named weighted lumping or real number lumping to extend the applicability of lumping to a more general class. We demonstrate sufficient conditions derived by integer lumping or weighted lumping in answer to three questions: 1. Given two HMMs of different orders, what are the criteria to determine their equivalence? 2. Given one HMM, how do we decide if there is a low-order realization and how do we construct a system of low-order that is dynamically consistent with the true system? 3. What is the uniqueness of realizations? We show that all types of lumpings are simply low-order realizations of non-minimal systems. Finally, we prove monotonicity of the approximation errors.

Chapter 2

Realizations of Non-minimal Systems

Given a true system, we are interested in characterizing the approximation error of HMMs that approximate the true system to a greatest extent. The first step to explore a best approximation is to examine the situation when the approximation error is zero, i.e. when one system is equivalent to the other. This belongs to the scope of realization theory, which is a special case of approximations. The general purpose of realization theory is to find a linear system with nonnegative entries in both its transition and emission matrices (called a *positive linear system*) to realize a given transfer function. HMMs are one special type of positive linear systems. In realization theory, the order of two HMMs could be the same or different. Particularly, we confine our attention to the situation when the order of a realization is less than the true system. Among the set of low-order candidate systems, we focus on realization questions such as: 1. Given two HMMs, how can we determine if one HMM is equivalent to the other? 2. What is the requirement for a HMM such that its low-order equivalent system exists? 3. If the low-order realization exists, is it unique? 4. How many low-order realizations are there for a HMM? 5. How can we construct low-order realizations from the true system with limited data? Note that no matter whether the approximation error is zero or not, this error does not depend on the learning algorithm or the amount of data available; it would simply be a measure of the distance between the true system and the set of candidate simple models.

When investigating the realizations, it is important to distinguish the class of systems that have low-order realizations from the rest of the candidate systems and characterize their behaviors and speciality. This is a minimality problem. The goal of a minimality problem is to identify the essential meaning of minimality in terms of order. One of our purposes is to calibrate the minimal requirement for a HMM that has a low-order equivalent realization. This is actually a difficult problem that remains unsolved because it is hard to factorize a positive linear system and classify all minimal realizations of positive linear systems. [26]. Understanding the notion of minimality will help us to interpret certain aggregation results of lumpable Markov processes found in the literature and to develop new equivalent results for such systems as alternate realizations of the same process.

In this section, we first characterize the meaning of distance, dynamic consistency, realization and best approximation. We then define minimality and explain its relationship with irreducibility and observability. We prove that a system is minimal if it is observable. After characterizing minimality, we point out that it is the linearly positiveness and stochasticity that make the realization problem difficult to solve. In order to reveal that best approximations with order greater or equal to the order of a minimal realization are actually realizations of the true system, we construct realizations by embedding a true system into high-order equivalent systems. For HMMs, we illustrate integer lumping and weighted lumping theory to find low-order realizations of non-minimal systems. We present three types of integer lumping and discuss their weaknesses. We further provide an approach named weighted lumping or real number lumping to extend the applicability of lumping to a more general class. We demonstrate sufficient and necessary conditions derived by integer lumping or weighted lumping in answer to three questions: 1. Given two HMMs of different orders, what are the criteria to determine their equivalence? 2. Given one HMM, how do we decide if there is a low-order realization and how do we construct a system of low-order that is dynamically consistent with the true system? 3. What is the uniqueness of realizations? Finally, we show that all types of lumpings are simply low-order realizations of non-minimal systems.

2.1 Low-order Realizations of Non-minimal Systems

Given a true system and two HMMs, how do we decide which HMM is closer to the true system than the other? In other words, how do we quantify the difference between systems? In order to measure the modeling errors between systems, we first introduce the notion of distance.

Definition 3 Given a true system (A, C) of order n,

$$x(k+1) = Ax(k)$$
$$y(k) = Cx(k)$$

and another system $(\widetilde{A}, \widetilde{C})$ of order $m < n$,

$$\tilde{x}(k+1) = \widetilde{A}\tilde{x}(k)$$
$$\tilde{y}(k) = \widetilde{C}\tilde{x}(k),$$

the *distance* between (A, C) and $(\tilde{A}, \tilde{C})$ is defined as:

$$d((A, C), (\tilde{A}, \tilde{C})) = \parallel y(k) - \tilde{y}(k) \parallel_2 \tag{2.1}$$

By this definition, the distance between two HMMs is measured by the difference between their output distributions in the form of a $2 - norm$. Since it does not matter what norm is chosen when the distance is zero for realizations (any norm on zero still gives a result of zero), we will explain more about the norm selection in the next chapter.

Definition 4 Given two systems (A, C) and $(\tilde{A}, \tilde{C})$, $(\tilde{A}, \tilde{C})$ is said to be *dynamically consistent* with (A, C) if $d((A, C), (\tilde{A}, \tilde{C})) = 0$.

Dynamic consistency is a property that suggests the equivalency of two systems. $(\tilde{A}, \tilde{C})$ is a *realization* of (A, C) if the distance between the two systems is zero; $(\tilde{A}, \tilde{C})$ is an *approximation* of (A, C) if the distance is nonzero; Among all the systems that have nonzero distance to the true system (A, C), $(\tilde{A}, \tilde{C})$ is a *best approximation* of (A, C) if the distance from $(\tilde{A}, \tilde{C})$ to (A, C) is minimal, i.e.

$$\min_{(\tilde{A},\tilde{C})\in S} d((A, C), (\tilde{A}, \tilde{C})) = \min_{(\tilde{A},\tilde{C})\in S} \min_{\tilde{x}_o} \max_{x_o} \parallel y(k) - \tilde{y}(k) \parallel_2 \tag{2.2}$$

In this equation, $\max_{x_0}$ means to start with the worst case of an initial condition in the true system to make the distance greatest; $\min_{\tilde{x}_0}$ means to search an initial condition among every possible initial condition in the state space of all low-order candidate systems to minimize the distance. Note that we confine our focus to a stochastic framework, in which all possible initial conditions $x_o \in \mathbb{R}^n_+$ and $\tilde{x}_o \in \mathbb{R}^n_+$ must be column stochastic.

These definitions demonstrate that, if a system of order less than the true system can yield an output distribution that minimizes the distance to the true system with an appropriate initial condition among all the possible initial conditions, this system is regarded as a best approximation of the true system. In terms of HMMs, two systems are equivalent if they have the same output. For example, if we have two casino machines, one with only one die and the other with 100 dice, it is really difficult for a player to realize the difference between the two machines if he observes two identical distribution sequences of points generated by the machines at each and every roll.

Definition 5 A system is said to be a *minimal system* if there is no system of low-order that has a zero distance to this system.

If a system is a minimal system, approximation errors will always exist between all the low-order systems and this true system. Minimality is a property that describes the minimal requirement of a system's order to guarantee the existence of a low-order realization of the true system. If a system has an equivalent system of order less than it, this system is a *non-minimal* system. To characterize minimality, we introduce the notion of reducibility and observability.

Definition 6 A matrix $A \in \mathbb{R}^{n\times n}$ is said to be *reducible* if either

(a) $n = 1$ and $A = 0$; or

(b) $n \geq 2$, there is a permutation matrix $P \in \mathbb{R}^{n\times n}$, and there is some integer r with $1 \leq r \leq n-1$ such that $P^T AP = \begin{bmatrix} B & C \\ 0 & D \end{bmatrix}$,

where $B \in \mathbb{R}^{r\times r}, D \in \mathbb{R}^{(n-r)\times(n-r)}, C \in \mathbb{R}^{r\times(n-r)}$ and $0 \in \mathbb{R}^{(n-r)\times r}$ is a zero matrix [18].

A matrix $A \in \mathbb{R}^{n\times n}$ is said to be *irreducible* if it is not reducible [18].

Definition 7 Given a system (A, C) of order n,

$$x(k+1) = Ax(k)$$

$$y(k) = Cx(k)$$

it is *observable* if the observable matrix O has full rank, where

$$O = \begin{bmatrix} C & CA & CA^2 & \dots & CA^{n-1} \end{bmatrix}^T$$

Theorem 2.1.1 If (A, C) of order n is observable, it is a minimal system.

Proof We prove this by contradiction. Suppose (A, C) is observable, but not minimal. Then for this non-minimal system (A, C), $\exists(\tilde{A}, \tilde{C})$ such that

$$d((A, C), (\tilde{A}, \tilde{C})) = 0.$$

This means the output distributions should also be equal at all times, that is, $\forall x(0)$,$\forall n$, $\exists \tilde{x}(0)$, such that

$$\begin{bmatrix} C \\ CA \\ \dots \\ CA^N \end{bmatrix} x(0) = \begin{bmatrix} \tilde{C} \\ \tilde{C}\tilde{A} \\ \dots \\ \tilde{C}\tilde{A}^N \end{bmatrix} \tilde{x}(0)$$

However $\forall N > n$, since (A, C) is observable,

for the left hand side, $dim(span \begin{bmatrix} C & CA & CA^2 & \dots & CA^N \end{bmatrix}^T) = n$;

for the right hand side, $dim(span \begin{bmatrix} \tilde{C} & \tilde{C}\tilde{A} & \tilde{C}\tilde{A}^2 & \dots & \tilde{C}\tilde{A}^N \end{bmatrix}^T) = m < n$

This results in a conflict between the dimension of the matrices on the left and right hand side. Therefore, (A, C) has to be minimal if it is observable.

□

This theorem indicates that observability is an important property to characterize minimality. If a system is non-minimal, it must be unobservable.

Hof also mentioned in [40] that if a positive linear system is observable and reachable, then the system is a minimal positive realization of it impulse response. Since we focus on autonomous systems with zero input, we do not need to deal with reachability. Topics related to observability can also be found in [41].

It is recognized that the realization problem, especially the problem of characterizing minimality of the state space for HMMs, is difficult due to the fact that they are positive stochastic processes. Partial solutions to the minimality problem were provided: [42] and [43] used geometric interpretation to attempt to find a complete solution, [25] and [44] constructed Hankel matrices to describe positive factorizability, and [45] employed primitiveness to solve the positive realization problem. Linear algebra can provide a realization to the true system easily by change of basis [46], but after linear transformation it is not easy to maintain the entries of probability in a range of $[0, 1]$ and to keep values in each column sum to one; not to mention to preserve the Markovian property if we want to construct an equivalent system from the true system. In addition, research usually focuses on finding a realization of equal or lesser order than the true system rather than a realization of higher order. We can construct a high-order equivalent realization by expanding it as follows:

Given a true system (A, C) of order n, construct a pair $(\bar{A}, \bar{C})$ of order $N > n$, where

$$\bar{A} = \begin{bmatrix} A_{n\times n} & 0_{n\times(N-n)} \\ 0_{(N-n)\times n} & I_{N-n} \end{bmatrix} \quad \bar{C} = \begin{bmatrix} C_{2\times n} & D_{2\times(N-n)} \end{bmatrix} \quad \bar{x}(0) = \begin{bmatrix} x_{n\times 1} \\ 0_{(N-n)\times 1}(0) \end{bmatrix}$$

and $D_{2\times(N-n)}$ is an arbitrary stochastic matrix.

We can verify that $y(k) = \bar{y}(k) = CA^k x(0)$. This shows that we can always find a high-order realization of a true system when the order of candidate systems is overestimated. What interests us more is to explore the realization problem within a set of low-order systems. Though there is no complete solution to find low-order realizations for an arbitrary positive linear system, there are methods that are able to generate a realization if we constrain the true system within a specific set. Lumping theory is one of them. Lumping is a special kind of aggregation in terms of preserving the Markovian property and stochasticity in the lower system. Generally speaking, lumpability means to partition the atomic states of a Markov chain into coarse groups which behave in a dynamically similar manner as the original system [37]. There are various types of lumping according to different classifications: we can classify lumping into integer lumping, weighted lumping based on the type of partition we use. We can classify lumping into homogeneous (stationary) lumping and inhomogeneous lumping according to the time variance property of the transition matrix. Stationary lumping can be further classified into subcategories according to the direction of the transition arrows in the automata. Most of the previous works focus on integer lumping. Integer Lumping has been exploited in various areas including speech recognition [47], system specification [38], and communication signal processing [48]. Kemeny addressed the concept of weak lumpability in [35]. More recently, Ledoux discussed the impact of various initial conditions on different forms of state aggregation in [49]. Most of the lumpability work is concerned with Markov chains and only recently has the concept been introduced to describe dynamics of HMMs [37]. Dogancy implicitly used integer lumping to find approximations of HMMs in [32]. White et al. further investigated the concept of integer lumping by separating the time scale in the Markov chain dynamics and achieved reductions of states by eliminating the "non-slow" states in [37].

To explain how lumping theory can help us to find realizations, we first provide a review on what has been done with lumping theory. The most common type of lumping is integer lumping, which only has integer entries in the partition matrix. Then we show the limitations of integer lumping in searching for a realization. After that, we present our weighted lumping approach which generalizes lumping theory.

2.1.1 Integer Lumping

Let $x_i(k) \in \mathbb{R}_+$ denote the i^{th} state in the state vector $x(k)$. Also let $S_x = \{s_1, \ldots, s_n\}$ denote the set of nodes in the corresponding automaton where $s_i^{(k)}$ represents the i^{th} node associated with $x_i(k)$.

Definition 8 [1] For an positive integer $m < n$, a Markov chain is said to be *weakly m-lumpable* if and only if $\exists\ \bar{S}_x = \{\bar{S}_1, \bar{S}_2, \ldots, \bar{S_m}\}$ where $\bar{S}_i \subset S_x, i \in \{1, \ldots, m\}$, $\bar{S}_i \cap \bar{S}_j = \emptyset$ if $i \neq j$ and $\bigcup_{i=1}^{m} \bar{S}_i = S_x$, such that $\forall i = 1, \ldots, m$, $Pr[s_r^{(k)} \in \bar{S}_i | s_p^{(k-1)} = h]$ is independent of $h, \forall h \in \bar{S}_j, \forall j \neq i$.

Weakly lumpability indicates that an aggregated transition probability from a node s_p in cluster $\bar{S}_j$ to another node s_r in cluster $\bar{S}_i$ $(i \neq j)$ does not depend on the

[1] A similar definition can be found in [36].

previous state where it stayed. In other words, if the high order system as well as the low order system are both Markov processes, weakly m-lumpability is preserved.

Definition 9 $L \in \mathbb{R}^{m \times n}$ is said to be an *aggregation operator* if it satisfies: $\mathcal{R}^n \rightarrow \mathcal{R}^m$, $l_{ji} = 1$ if $s_i \in \bar{S}_j$ and $l_{ji} = 0$ otherwise.

With a particular partition, each node in the automaton will be grouped into one unique cluster. l_{ij} serves as an indicator demonstrating whether node i is grouped in cluster $\bar{S}_j$ or not.

Lemma 2.1.2 If $L \in \mathbb{R}^{m \times n}$*,* $m < n$*,* then $rank(L) = m$*.*

Proof With rank inequality, $rank(L) \leq m$. According to the definition of aggregation operator, $l_{ij} \in \{0, 1\}$. Since there is no empty cluster in the aggregation, each row must have at least one 1, and each column can only have one 1; in order to keep stochastic property, each column can only have one 1. Therefore, there are m linear independent row vectors in L, i.e. $rank(L) = m$.

□

An aggregation operator L enables us to partition the true system and group n states into m clusters. It will relate the high and low state vectors in a coherent steady way.

Definition 10 A low-order system is said to be *state dynamically consistent* with the true system if there is an aggregation operator L such that $\forall k$,

$$\tilde{x}(k) = Lx(k) \tag{2.3}$$

This property suggests that once L and initial conditions are fixed, behaviors of the low-order state vector will be determined by the high-order state vector.

With an aggregation operator, it is possible for lumpings to generate different low-order realizations. According to the time variance property and the direction of transition probabilities, there are basically three types of integer lumping for a Markov process: general lumping, ordinary lumping and exact lumping.

Let g, o, e denote the type of integer lumping: general, ordinary and exact. In lumped models, denote $\tilde{A}_t^m(k) \in \mathbb{R}^{m \times m}$ as an m^{th} order transition matrix, using one of the lumping types $t = \{g, o, e\}$. Let $\tilde{a}_{pq}(k)$ denote an transition probability from cluster $\bar{S}_q$ to $\bar{S}_p$ at time k in $\tilde{A}_t^m(k)$.

Definition 11 A lumping is said to be a *general lumping* if it satisfies:
(i) $\tilde{x}(k) = \sum_{i:s_i \in \bar{S}_j} x(k), i = 1, \ldots, n, j = 1, \ldots, m;$
(ii) $\tilde{a}_{pq}(k) = \sum_{j:s_j \in \bar{S}_q} \lambda_{jq}(k) \sum_{i:s_i \in \bar{S}_p} a_{ij}$, where $\lambda_{jq}(k) = Pr[s_j | s_h \in \bar{S}_q]$

General lumpability is a time variant exact reduction of the higher order model. Condition (i) indicates that the value of a new aggregated state in a low-order model is the sum of all the state values within that aggregated group in the original model. Though there are other ways to aggregate the states in the vector, they do not necessarily preserve stochasticity. A sum operation ensures stochasticity. Condition (ii) calculates the transition probability matrix in the low-order model (its derivation can be found in the Appendix 5.2).

Example 2.1 If we have a 3rd order column stochastic Markov chain:

$$\begin{bmatrix} x1(k+1) \\ x2(k+1) \\ x3(k+1) \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x1(k) \\ x2(k) \\ x3(k) \end{bmatrix}$$

We want to reduce it to a 2nd order model by aggregating the first two states into one cluster and leave the third one as the other cluster. Compute the transition probabilities according to condition (ii),
$\tilde{x}_1(k) = x_1(k) + x_2(k)$; $\tilde{x}_2(k) = x_3(k)$;

$\tilde{a}_{11}(k) = \frac{x_1(k)}{x_1(k)+x_2(k)} \cdot (a_{11} + a_{21}) + \frac{x_2(k)}{x_1(k)+x_2(k)} \cdot (a_{12} + a_{22})$;

$\tilde{a}_{21}(k) = \frac{x_1(k)}{x_1(k)+x_2(k)} \cdot a_{31} + \frac{x_2(k)}{x_1(k)+x_2(k)} \cdot a_{32}$;

$\tilde{a}_{12}(k) = a_{13} + a_{23}$; $\tilde{a}_{22}(k) = a_{33}$.

It can be verified that

$$\forall k, \begin{bmatrix} \tilde{x}_1(k+1) \\ \tilde{x}_2(k+1) \end{bmatrix} = \tilde{A}_g^{(2)}(k) \cdot \begin{bmatrix} \tilde{x}_1(k) \\ \tilde{x}_2(k) \end{bmatrix}$$

□

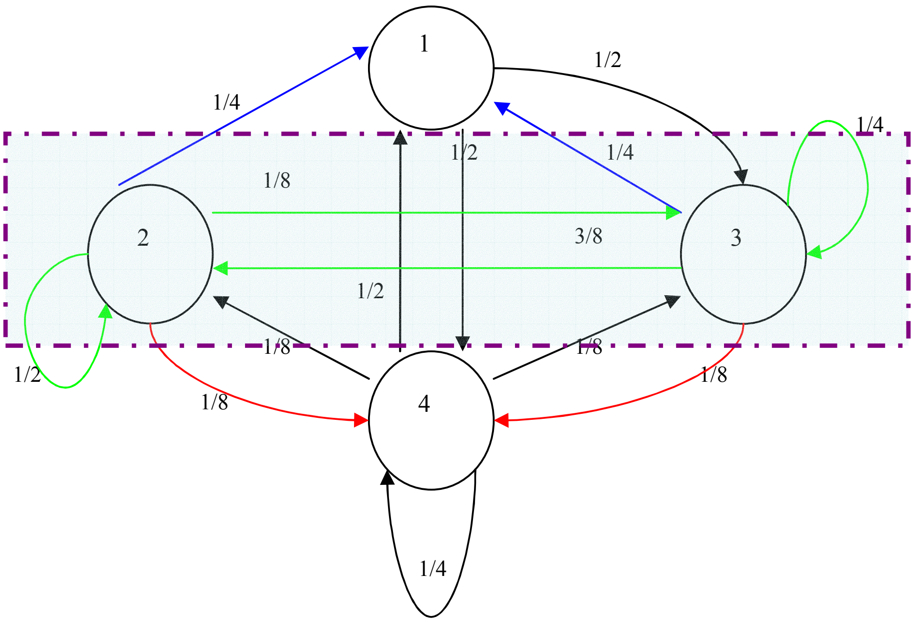
With a general lumping, the evolution of a reduced model is given by:

$$\tilde{x}(k+1) = \tilde{A}_g^m(k)\tilde{x}(k) = \tilde{A}_g^m(k)\tilde{A}_g^m(k-1)\tilde{x}(k-1) = \ldots = \prod_{i=0}^{k} \tilde{A}_g^m(i)\tilde{x}(0) \tag{2.4}$$

The capability of general lumping to preserve dynamic consistency stems from the fact that it turns a homogeneous system into an inhomogeneous system of small dimension. Though the size is reduced, the transition matrix now becomes a function of time. Each entry in $\tilde{A}_g^m(k)$ is refreshed based on non-stationary conditional probabilities calculated from pairs of starting and ending states at each round of transition. Indeed, this approach does not save computational expenditure because a large amount of memory is sacrificed to store updated values at each round for the benefit of reduction.

General lumping is not desirable since it produces non-stationary systems of low-order, but it enjoys the strength of having no requirements for the transition matrix. The following two integer lumping approaches have the advantage of generating homogeneous systems and saving computation. However, not all Markov chains meet the corresponding lumpable requirements.

Definition 12 A lumping is said to be an *ordinary lumping* if it satisfies:



The arrows with the same color other than black have the same transition probability.

Figure 2.1: **An automaton of the ordinary lumping example**

$(i)\ \tilde{x}(k) = \sum\limits_{i:s_i \in \bar{S}_j} x(k), i = 1, \ldots, n, j = 1, \ldots, m;$

$(ii)\ \tilde{a}_{pq} = \sum\limits_{h:s_h \in \bar{S}_p} a_{hi}, i : s_i \in \bar{S}_p.$

(i) is the same as the first condition in general lumping. (ii) indicates that ordinary lumping conducts aggregation by considering all the outgoing directions of the transition probabilities. If two nodes i, j transit out to a third node h in $\bar{S}_p$ with the same probability $\mathbf{p}$, i and j can be gathered into one cluster $\bar{S}_q$. The probability from $\bar{S}_q$ to h is thus $\mathbf{p}$, i.e, if $Pr[s_h^{(k+1)}|s_i^{(k)}] = Pr[s_h^{(k+1)}|s_j^{(k)}] = \mathbf{p}$, then $\tilde{a}_{pq} = \mathbf{p}$. We can also treat ordinary lumping as this: when the outgoing probabilities are equal, the value of $\lambda_{jq}(k)$ stays the same for all k due to fact that the ratio of the original state value to the aggregated state value is fixed. In this sense, ordinary lumping is a special case of general lumping.

Most of the previous work on ordinary lumping used the definition to compute the low-order transition matrix $\widetilde{A}$. This is sometimes inconvenient. However, if we know a matrix A is ordinary lumpable and the probabilities between the aggregated nodes are row-wise uniformly distributed, other than using the definition, we found an easy way to calculate $\widetilde{A}$.

Theorem 2.1.3 If A is ordinary lumpable by aggregating states $x_p, x_{p+1}, \ldots, x_q$, $1 \leq p < q \leq n$, denote $r = q - p + 1$; $a_{ip} = a_{i(p+1)} = \ldots = a_{iq}$, $i = \{p, p+1, \ldots, q\}$; then $\exists \widetilde{A} = LAR$ such that $\forall k$, $\forall x(0)$, $\tilde{x}(k) = Lx(k)$ holds, where $R \in \mathbb{R}_{+}^{n \times m}$ denote the pseudoinverse of L and $LR = I_{m \times m}$.

Proof Without loss of generality, we assume that we lump a nth order ordinary lumpable model into a m^{th} order model by aggregating states $x_1, x_2, \ldots, x_r$, $r = n - m + 1$. Denote v_i as the value of a_{ip}, i=1,. . . ,r. The transition matrix A should look like:

$$A = \begin{bmatrix} v_1 & \ldots & v_1 & a_{1(r+1)} & \ldots & a_{1n} \\ v_2 & \ldots & v_2 & a_{2(r+1)} & \ldots & a_{1n} \\ v_r & \ldots & p_r & a_{r(r+1)} & \ldots & a_{rn} \\ a_{(r+1)1} & \ldots & \ldots & \ldots & \ldots & a_{(r+1)n} \\ a_{n1} & \ldots & \ldots & \ldots & \ldots & a_{nn} \end{bmatrix}$$

Then the aggregation operator $L \in \mathbb{R}_{+}^{m \times n}$ and its pseudoinverse $R \in \mathbb{R}_{+}^{n \times m}$ should look like:

$$L = \begin{bmatrix} 1 & \ldots & 1 & 0 \\ 0 & \ldots & 0 & I_{(m-1),(m-1)} \end{bmatrix} \quad R = \begin{bmatrix} \frac{1}{r} & 0 & \ldots & 0 \\ \ldots & \ldots & \ldots & \\ \frac{1}{r} & 0 & \ldots & 0 \\ 0 & & I_{(m-1),(m-1)} & \end{bmatrix}$$

In the first row of L, there are $(n - m + 1)$ ones; the submatrix from $l_{2(r+1)}$ to l_{mn}

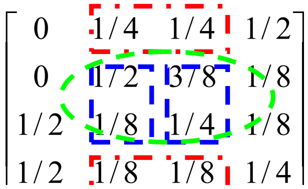


Figure 2.2: **A matrix representation of the ordinary lumping example**

is a $(m-1)\times(m-1)$ identity matrix. Correspondingly, in the first column of R, there are $(n-m+1)$ $\frac{1}{r}$ and the submatrix is a $(m-1)\times(m-1)$ identity matrix.

$$ARL = A\begin{bmatrix} \frac{1}{r} & \dots & \frac{1}{r} & 0 & \dots & 0 \\ & \ddots & & \vdots & \ddots & \vdots \\ \frac{1}{r} & \dots & \frac{1}{r} & 0 & \dots & 0 \\ 0 & \dots & 0 & & I_{(m-1),(m-1)} & \end{bmatrix} = A$$

$$\text{Thus,}\forall k,\ \widetilde{x}(k) = Lx(k) = LA^k x(0) = L(ARL)^k x(0) = LARLAR\dots ARLx(0)$$
$$= (LAR)\dots(LAR)\widetilde{x}(0) = (LAR)^k\widetilde{x}(0) = \widetilde{A}^k\widetilde{x}(0)$$

Therefore, instead of using the definition, we can directly construct a m^{th} order matrix by $\widetilde{A} = LAR$ and keep the state dynamic consistency.

Example 2.2 See Figure 2.1 and Figure 2.2. $A \in \mathbb{R}^{4\times 4}$:

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \\ x_4(k+1) \end{bmatrix} = \begin{bmatrix} 0 & 1/4 & 1/4 & 1/2 \\ 0 & 1/2 & 3/8 & 1/8 \\ 1/2 & 1/8 & 1/4 & 1/8 \\ 1/2 & 1/8 & 1/8 & 1/4 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ x_4(k) \end{bmatrix}$$

Since $a_{12} = a_{13} = 1/4$, $a_{22} + a_{32} = 1/2 + 1/8 = a_{23} + a_{33} = 3/8 + 1/4$ and $a_{42} = a_{43} = 1/8$, A is ordinary lumpable. We use ordinary lumping to aggregate state

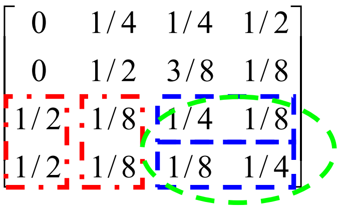


Figure 2.3: **A matrix representation of the exact lumping example**

$x_2(k)$ and $x_3(k)$ into one cluster $\widetilde{x}_{o2}$. The ordinary lumped model is:

$$\begin{bmatrix} \widetilde{x}_{o1}(k+1) \\ \widetilde{x}_{o2}(k+1) \\ \widetilde{x}_{o3}(k+1) \end{bmatrix} = \begin{bmatrix} 0 & 1/4 & 1/2 \\ 1/2 & 5/8 & 1/4 \\ 1/2 & 1/8 & 1/4 \end{bmatrix} \begin{bmatrix} \widetilde{x}_{o1}(k) \\ \widetilde{x}_{o2}(k) \\ \widetilde{x}_{o3}(k) \end{bmatrix}, \qquad \text{where } L_o = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

□

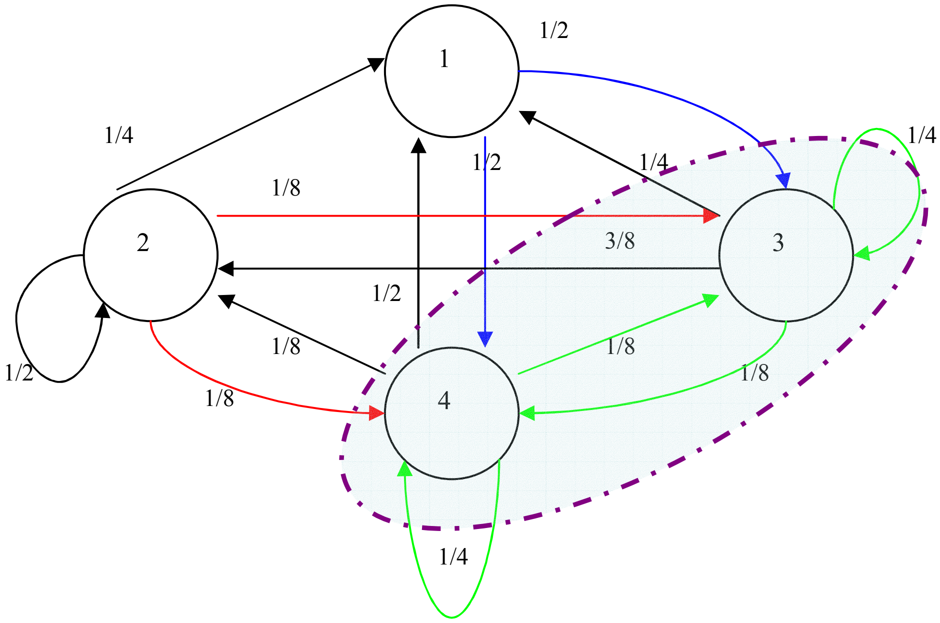
Definition 13 Let $r(i)$ denote the number of states in cluster $\bar{S}_i$ and v_i denote the value of each state in cluster $\bar{S}_i$. A lumping is said to be an *exact lumping* if it satisfies:

(i) $\tilde{x}(k) = \sum\limits_{i:s_i \in \bar{S}_j} x(k) = r(i) \cdot v_i, i = 1, \ldots, n, j = 1, \ldots, m;$

(ii) $\tilde{a}_{pq} = \frac{r(\bar{S}_p)}{r(\bar{S}_q)} \sum\limits_{j:s_j \in \bar{S}_q} a_{ij}, i : s_i \in \bar{S}_p, q = 1, \ldots, m$

(i) demonstrates that states within each cluster have to be uniformly distributed. (ii) indicates that exact lumping achieves an aggregation by considering incoming transition probabilities. If a third individual node transfers to node i and j with equal probability and the inter-transitional probabilities between i and j are the same, we aggregate them into one cluster.

Example 2.3 We still use the true system in Example 2.2. Since $a_{31} = a_{41} = 1/2$, $a_{32} = a_{42} = 1/8$ and $a_{33} + a_{34} = 1/4 + 1/8 = a_{43} + a_{44} = 1/8 + 1/4$, A is exact



The arrows with the same color other than black have the same transition probability.

Figure 2.4: **An automaton of the exact lumping example**

lumpable. By exact lumping, we aggregate state $x_3(k)$ and $x_4(k)$ as one cluster $\widetilde{x}_{e3}$ (see Figure 2.4 and Figure 2.3). Using L_e, the exact lumped model can be represented as:

$$\begin{bmatrix} \widetilde{x}_{e1}(k+1) \\ \widetilde{x}_{e2}(k+1) \\ \widetilde{x}_{e3}(k+1) \end{bmatrix} = \begin{bmatrix} 0 & 1/4 & 3/8 \\ 0 & 1/2 & 1/4 \\ 1 & 1/4 & 3/8 \end{bmatrix} \begin{bmatrix} \widetilde{x}_{e1}(k) \\ \widetilde{x}_{e2}(k) \\ \widetilde{x}_{e3}(k) \end{bmatrix}, \qquad \text{where } L_e = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

Note that even though the true system is the same, this lower system is different from the one we obtained from ordinary lumping in the previous example.

□

Lemma 2.1.4 Ordinary and exact lumping are the only two ways to achieve a low-order homogeneous system if integer lumping is employed to obtain a realization.

Proof This can be proved by contradiction. Suppose there is a third kind of integer lumping other than ordinary or exact lumping that can produce a low-order time invariant system and keep state dynamic consistency. For the sake of easiness, we show the simplest case of reducing a third-order system to a time invariant second-order system. Given we have a random transition matrix

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

Also assume that we want to aggregate states $x_1(k)$ and $x_2(k)$. (Situations of aggregating other states can be proved similarly). The aggregation operator L is:

$$L = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

In order to obtain a realization with integer lumping, the relationship $\widetilde{x}(k) = Lx(k)$ must hold. Thus, $\widetilde{x}_1(k) = x_1(k) + x_2(k)$ and $\widetilde{x}_3(k) = x_3(k)$, $\forall k$. Rewrite this relationship as:

$$\widetilde{A}^k \widetilde{x}(0) = LA^k x(0),\ \forall k,\ \forall x(0)$$

$$\Longleftrightarrow \begin{bmatrix} \widetilde{a}_{11} & \widetilde{a}_{12} \\ \widetilde{a}_{21} & \widetilde{a}_{22} \end{bmatrix} \begin{bmatrix} x_1(0) + x_2(0) \\ x_3(0) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x_1(0) \\ x_2(0) \\ x_3(0) \end{bmatrix}, \forall k, \forall x(0)$$

$$\Longleftrightarrow \begin{bmatrix} \widetilde{a}_{11} & \widetilde{a}_{12} \\ \widetilde{a}_{21} & \widetilde{a}_{22} \end{bmatrix} \begin{bmatrix} x_1(0) + x_2(0) \\ x_3(0) \end{bmatrix} = \begin{bmatrix} a_{11} + a_{12} & a_{12} + a_{22} & a_{13} + a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x_1(0) \\ x_2(0) \\ x_3(0) \end{bmatrix}, \forall k, \forall x(0)$$

$$\begin{bmatrix} \widetilde{a}_{11}(x_1(0) + x_2(0)) + \widetilde{a}_{12} x_3(0) \\ \widetilde{a}_{21}(x_1(0) + x_2(0)) + \widetilde{a}_{22} x_3(0) \end{bmatrix} = \begin{bmatrix} (a_{11} + a_{12})x_1(0) + (a_{12} + a_{22})x_2(0) + (a_{13} + a_{23})x_3(0) \\ a_{31}x_1(0) + a_{32}x_2(0) + a_{33}x_3(0) \end{bmatrix}$$

Because this should work for all k and $x(0)$, the following must be true:

$$\widetilde{a}_{11} = a_{11} + a_{21} = a_{12} + a_{22}, \quad \widetilde{a}_{12} = a_{13} + a_{23}$$
$$\widetilde{a}_{21} = a_{31} = a_{32}, \qquad \widetilde{a}_{22} = a_{33}$$

However, the above four equations are actually the conditions required by ordinary lumping or exact lumping. It raises a conflict against our assumption that this type of lumping is not ordinary or exact lumping. Therefore, a third type of integer lumping does not exist to produce a low-order realization.

□

We can also understand this lemma by thinking in this way: stationary integer lumpings simply consider the direction of transition probabilities when there are equal transition probabilities. A node in an automaton can merely have two directions of transition: incoming and outgoing. Ordinary lumping deals with the outgoing probabilities and exact lumping considers the incoming probabilities. Thus, they are the only two stationary integer lumpings to find a realization.

We point out that integer lumping has some interesting properties such as transitivity and strictly confluency. These properties describe how repeatable integer lumping can be conducted on the lumped system. However, we do not explain these properties in the thesis. Further information can be found in [38], which proved that general lumping and ordinary lumping are transitive, but exact lumping is not.

2.1.2 Problems with Lumping

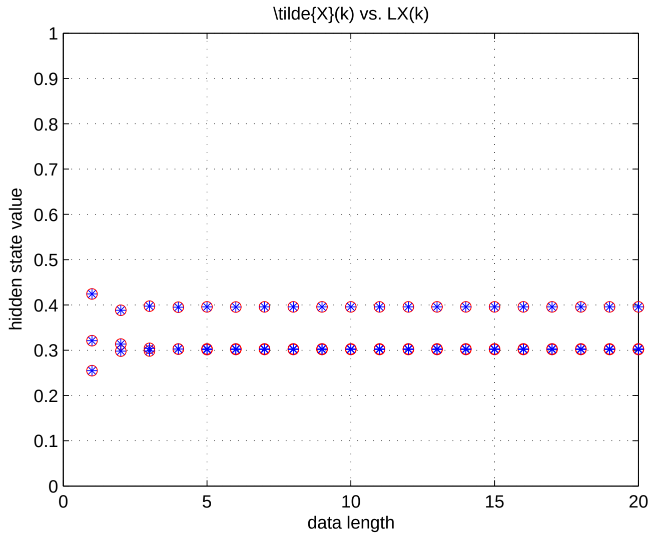
The previous section shows that integer lumping can preserve a state dynamic consistency between two HMMs and thus is likely to generate an equivalent low-order realization of the true system if the true system is lumpable. However, when choosing the type of lumping, we seem to prefer time invariant systems although general lumping is good at producing a low-order time variant system given an arbitrary transition matrix. This is because the estimated model obtained from the learning algorithm is also time invariant. To measure the modeling error between a realization and an estimation, it is desirable to compare two time invariant HMMs instead of a time variant system and a time invariant system. As we mentioned before, a pair (A, C) can represent a HMM and be regarded as a point in the plane of a set of systems. On one hand, a time variant transition matrix implies that the point is moving all the time. On the other hand, a time invariant estimate corresponds to a fixed point. It is difficult to measure the distance between two points if one of the points is wiggling continuously. In this sense, stationary integer lumping are favored over general lumping. However, there are three problems with stationary integer lumpings.

First, integer lumping is set specific. This means that, even if a true system is ordinary lumpable or exact lumpable, it can only be aggregated into certain orders. Moreover, these achievable orders may not be consecutive. It is possible that some true system of order n can be reduced to a simpler time invariant system of order m_2 but cannot be reduced to time invariant systems of order m_1 and m_3 where $m_1 < m_2 < m_3$.

Example 2.4 Given

$$A = \begin{bmatrix} 1/10 & 1/20 & 1/5 & 1/16 & 1/4 \\ 1/10 & 2/20 & 1/20 & 2/16 & 0 \\ 1/10 & 3/20 & 1/20 & 1/16 & 1/8 \\ 4/10 & 4/10 & 4/10 & 5/16 & 1/2 \\ 3/10 & 3/10 & 3/10 & 7/16 & 1/8 \end{bmatrix} \qquad C = \begin{bmatrix} 0.1 & 0.1 & 0.1 & 0.8 & 0.4 \\ 0.9 & 0.9 & 0.9 & 0.2 & 0.6 \end{bmatrix}$$

Through ordinary lumping, we can get a $\tilde{A}_o^3 \in \mathbb{R}^{3\times 3}$ and $\tilde{C}_o^3 \in \mathbb{R}^{2\times 3}$ by aggregating the first three states using L:



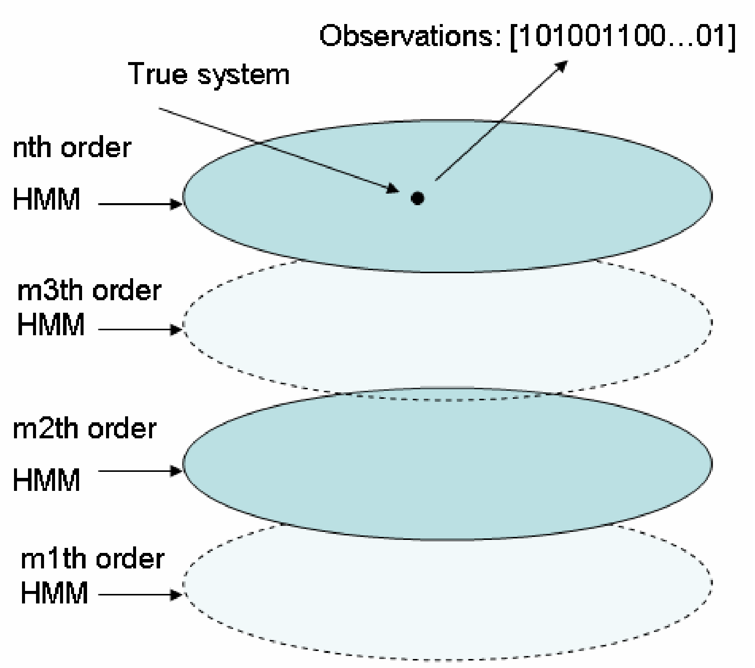
Red circles represent the value of *Lx(k)* in the fifth order system; blue stars represent the value of $\tilde{x}(k)$ in the third order system. At different time step *k*, red circles and blue stars always overlap with each other, which suggests that the two HMMs are state dynamically consistent with each other by an aggregation operator *L*.

Figure 2.5: **Output distributions of two HMMs**

$$L = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad \tilde{A}_o^3 = \begin{bmatrix} 3/10 & 1/4 & 3/8 \\ 4/10 & 5/16 & 1/2 \\ 3/10 & 7/10 & 1/8 \end{bmatrix} \quad C = \begin{bmatrix} 0.1 & 0.8 & 0.4 \\ 0.9 & 0.2 & 0.6 \end{bmatrix}$$

We can verify that, $\forall k$,$\forall x(0)$, $\tilde{x}(k) = Lx(k)$ and $\tilde{y}(k) = y(k)$. See Figure 2.5.

In this example, ordinary lumping can generate a 3^{rd} order realization; however, there is neither a 4^{th} order nor a 2^{nd} order time-invariant system that is able to keep dynamic consistency. Recall Lemma 2.1.4 in the previous section: with stationary integer lumping, the transition matrix must be either ordinary lumpable or exact lumpable in order to produce a time invariant low-order system. If a true system satisfies ordinary lumpability, the sum of each sub-column within a cluster must be the same. In the above example, the 5^{th} order system can be lumped to a 3^{rd} order system. This is so because its transition matrix A satisfies this requirement: $a_{11} + a_{21} + a_{31} = a_{12} + a_{22} + a_{32} = a_{13} + a_{23} + a_{33} = \frac{3}{10}$, $a_{41} = a_{42} + a_{43} = \frac{4}{10}$ and



Each plane represents a set of systems at a certain order. A solid circle refers to a set of systems that contains at least one time invariant dynamically consistent system through stationary integer lumping; while a dashed circle refers to a set that does not contain any of this kind of system. Solid circles and dashed circles would appear in various orders given different true systems.

Figure 2.6: **Systems of different orders**

$a_{51} = a_{52} + a_{53} = \frac{3}{10}$. Similarly, to get a low-order system through exact lumping, the sum of each sub-row within the corresponding clusters must be the same. In this example, A has none of the above property to be reduced to a system of order two or four. Thus, it is impossible to get a system of order 2 or 4 that is dynamically consistent with the true system.

□

This example indicates that for a fixed true system, there will not always be a series of adjacent time invariant lumped systems of different orders, but only in a certain order can a time invariant realization exist (also see Figure 2.6).

Second, integer lumping can be conducted only on some specific true systems. Not every true system can be lumped into a homogeneous low-order system. It is also possible that other operators can produce an output distribution the same as that of the true system, while integer lumping fails to generate a realization.

Example 2.5 Given

$$A = \begin{bmatrix} 1/3 & 1/4 & 1/5 \\ 1/3 & 1/2 & 2/5 \\ 1/3 & 1/4 & 2/5 \end{bmatrix} \qquad C = \begin{bmatrix} 7/12 & 1/2 & 17/24 \\ 5/12 & 1/2 & 7/24 \end{bmatrix}$$

No aggregation operator can reduce this 3^rd order model into a 2^{nd} order model (If we want a realization by aggregating the first and the third states in the true system, a_{13} must be equal to a_{33}; if we want a lumping by aggregating the second and the third states, a_{22} and a_{32} must be equal.) However, we can find a pair $(\widetilde{A}, \widetilde{C})$

$$\widetilde{A} = \begin{bmatrix} 793/1440 & 11743/27360 \\ 647/1440 & 15617/27360 \end{bmatrix} \qquad \widetilde{C} = \begin{bmatrix} 1/3 & 5/6 \\ 2/3 & 1/6 \end{bmatrix}$$

such that $\forall k$, $\tilde{x}(k+1) = \widetilde{A}\tilde{x}(k)$, $\tilde{y}(k) = \widetilde{C}\tilde{x}(k)$, and $\tilde{y}(k) = y(k)$.

□

Third, integer lumping is not a unique way to achieve a realization. When integer lumping can produce a realization with an integer aggregation operator, there could also be another operator that helps to generate the same output as integer lumping does.

Example 2.6 Given a 3^{rd} order true system

$$A = \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{bmatrix} \qquad C = \begin{bmatrix} 1/2 & 1/2 & 1/2 \\ 1/2 & 1/2 & 1/2 \end{bmatrix}$$

If we use an aggregation operator

$$L = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \text{ we get an } \tilde{A} = \begin{bmatrix} 2/3 & 1/3 \\ 2/3 & 1/3 \end{bmatrix}$$

that preserves $\tilde{x}(k) = Lx(k)$ and gives the same output distribution

$$\tilde{y}(k) = y(k) = \begin{bmatrix} 1/2 \\ 1/2 \end{bmatrix}.$$

But we can also have another lower system pair $(\widetilde{A}, \widetilde{C})$ that can not be computed from any aggregation operator:

$$\widetilde{A} = \begin{bmatrix} 1/2 & 1/2 \\ 1/2 & 1/2 \end{bmatrix} \qquad \widetilde{C} = \begin{bmatrix} 1/2 & 1/2 \\ 1/2 & 1/2 \end{bmatrix}$$

such that $\forall k, \forall x(0)$, $\tilde{y}(k) = y(k)$ holds (We cannot obtain $(\widetilde{A}, \widetilde{C})$ from any integer aggregation operator because if there is an L' that can produce $\widetilde{A}$, then there should be an entry of value $\frac{2}{3}$ in $\widetilde{A}$ that indicates aggregation from integer lumping). □

These examples show that integer lumping is not a unique way to generate an equivalent realization. In the situation where integer lumping cannot produce a desired system, we are still able to find a realization through other type of operators. In the following, we characterize another type of operator that can generate equivalent realizations as well.

2.1.3 Weighted Lumping

We explained in the previous section the weakness of integer lumping and showed that there could be some other operator that helps a low-order system to preserve dynamic consistency. In this section, we will introduce a new type of lumping and explain its relationship with low-order realizations.

Recall the previous two examples. In Example 2.6,

$$A = \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{bmatrix} \qquad C = \begin{bmatrix} 1/2 & 1/2 & 1/2 \\ 1/2 & 1/2 & 1/2 \end{bmatrix},$$

$$\widetilde{A} = \begin{bmatrix} 1/2 & 1/2 \\ 1/2 & 1/2 \end{bmatrix} \qquad \widetilde{C} = \begin{bmatrix} 1/2 & 1/2 \\ 1/2 & 1/2 \end{bmatrix}$$

there is another operator

$$\begin{bmatrix} 1 & 1/2 & 0 \\ 0 & 1/2 & 1 \end{bmatrix}$$

that helps the system preserve dynamic consistency. We verify this below:

$$\tilde{x}(k) = \begin{bmatrix} 1 & 1/2 & 0 \\ 0 & 1/2 & 1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix} = \widetilde{A}^k \tilde{x}(0) = \begin{bmatrix} x_1(0) + 1/2x_2(0) \\ 1/2x_2(0) + x_3(0) \end{bmatrix}$$

$$\tilde{y}(k) = \widetilde{C}\tilde{x}(k) = y(k) = Cx(k) = \begin{bmatrix} 1/2x_1(k) + 1/2x_2(k) + 1/2x_3(k) \\ 1/2x_1(k) + 1/2x_2(k) + 1/2x_3(k) \end{bmatrix}$$

In Example 2.5,

$$A = \begin{bmatrix} 1/3 & 1/4 & 1/5 \\ 1/3 & 1/2 & 2/5 \\ 1/3 & 1/4 & 2/5 \end{bmatrix} \qquad C = \begin{bmatrix} 7/12 & 1/2 & 17/24 \\ 5/12 & 1/2 & 7/24 \end{bmatrix}$$

$$\widetilde{A} = \begin{bmatrix} 793/1440 & 11743/27360 \\ 647/1440 & 15617/27360 \end{bmatrix} \qquad \widetilde{C} = \begin{bmatrix} 1/3 & 5/6 \\ 2/3 & 1/6 \end{bmatrix}$$

there is another operator

$$\begin{bmatrix} 1/2 & 2/3 & 1/4 \\ 1/2 & 1/3 & 3/4 \end{bmatrix}$$

that also helps the lower system to maintain dynamic consistency.

Notice that each entry of in the operator is a real number between $[0, 1]$. It is reasonable to extend the notion of integer lumping into real number lumping.

Definition 14 L_w is called a *weighted aggregation operator* if it is column stochastic, i.e.

$$L_w \in \mathbb{R}_+^{m\times n},\ 0 \leq l_{i,j} \leq 1,\ \sum_{i=1}^{m} l_{ij} = 1, j = 1, \ldots, n.$$

Definition 15 L_n is called a *non-weighted aggregation operator* if each column of it sums to 1, i.e.

$$L_n \in \mathbb{R}^{m\times n},\ \sum_{i=1}^{m} l_{ij} = 1, j = 1, \ldots, n.$$

Recall the definition of an integer aggregation operator: $L \in \mathbb{R}_+^{m\times n}$, $l_{i,j} \in \{0, 1\}$, $\sum_{i=1}^{m} l_{ij} = 1, j = 1, \ldots, n$. The difference between L_w and L is that each entry in L can be only either 0 or 1 while L_w can take any real number between $[0, 1]$. Lumping with a weighted aggregation operator is called *weighted lumping.* Lumping with a non-weighted aggregation operator is called *non-weighted lumping.* Integer Lumping is actually a very special case of weighted lumping. L_n further relaxes the constraints on the operator to include any real number, e.g., negative numbers or entries larger than 1.

Lemma 2.1.5 Given two column stochastic matrices $A \in \mathbb{R}^{n\times n}$ and $C \in \mathbb{R}^{m\times n}$, $W = CA \in \mathbb{R}^{m\times n}$ is a column stochastic matrix.

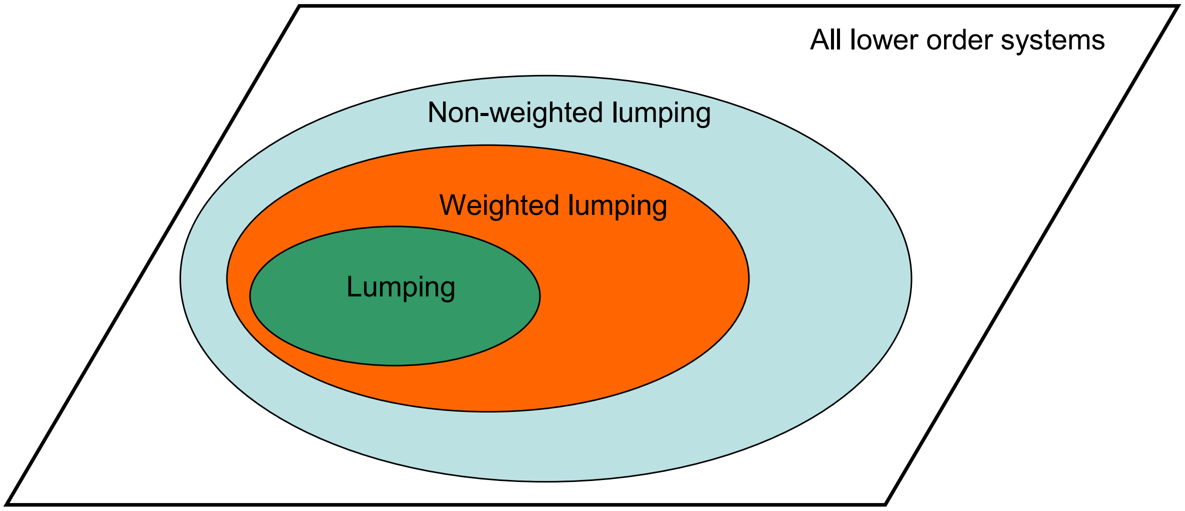


Figure 2.7: **Relationship of different lower order system sets**

Proof Given

$$A = \begin{bmatrix} a_{11} & \dots & a_{1n} \\ \dots & a_{jr} & \dots \\ a_{n1} & \dots & a_{nn} \end{bmatrix} \qquad C = \begin{bmatrix} c_{11} & \dots & c_{1n} \\ \dots & c_{ij} & \dots \\ c_{m1} & \dots & c_{mn} \end{bmatrix}$$

because A and C are column stochastic,

$$\sum_{j=1}^{n} a_{jr} = 1, r = 1, \dots n \qquad \sum_{i=1}^{m} c_{ij} = 1, j = 1, \dots, n.$$

For each column j in W, we have:

$w_{1j} = c_{11}a_{1j} + c_{12}a_{2j} + \dots + c_{1n}a_{nj}$,

$w_{2j} = c_{21}a_{1j} + c_{22}a_{2j} + \dots + c_{2n}a_{nj}$,

$\dots$,

$w_{mj} = c_{m1}a_{1j} + c_{m2}a_{2j} + \dots + c_{mn}a_{nj}$.

The sum of j^{th} column is:

$$\sum_{i=1}^{m} w_{ij} = a_{1j}\sum_{i=1}^{m} c_{i1} + a_{2j}\sum_{i=1}^{m} c_{i2} + \ldots + a_{nj}\sum_{i=1}^{m} c_{in} = a_{1j} + a_{2j} + \ldots + a_{nj} = 1.$$

Therefore, W is a column stochastic matrix.

□

We now have shown that the product of two column stochastic matrices has to be a column stochastic matrix. The theorem below captures the relationship between the hidden state vectors and the operator.

Denote $\mathbb{L}$ as an operator representing the relationship between $x(k)$ and $\tilde{x}(k)$. It could be lumping operator L, weighted lumping operator L_w or non-weighted lumping operator L_n.

Theorem 2.1.6 If for $\mathbb{L} \in \mathbb{R}^{m\times n}$, $x(k) \in \mathbb{R}^n_+$ is column stochastic, $\forall k, \forall x(0)$, $\tilde{x}(k) = \mathbb{L}x(k)$, then $\tilde{x}(k) \in \mathbb{R}^m$ is column stochastic if and only if $\mathbb{L}$ is column stochastic.

Proof Given $0 \le x_j(k) \le 1$, $\sum_{j=1}^{n} x_j(k) = 1$, $\tilde{x}(k) = \mathbb{L}x(k)$, it is equivalent to prove

$$\{0 \le l_{ij} \le 1 \cap \sum_{i=1}^{m} l_{ij} = 1, j = 1, 2, \ldots, n\} \iff \forall x(0), \{0 \le \tilde{x}(k) \le 1 \cap \sum_{i=1}^{m} \tilde{x}_i(k) = 1\}.$$

Sufficiency ($\Longrightarrow$)**:**

$\tilde{x}_i(k) = l_{i1}x_1(k) + l_{i2}x_2(k) + \ldots + l_{in}x_n(k) = \sum_{j=1}^{n} l_{ij}x_j(k).$

If $0 \le l_{ij} \le 1$, $0 \le x_j(k) \le 1$, then $0 \le l_{ij}x_j(k) \le x_j(k) \le 1 \Rightarrow 0 \le \tilde{x}_i(k) \le 1$ **(1)**

$$\sum_{i=1}^{m} \tilde{x}_i(k) = \sum_{j=1}^{n} x_j(k) \sum_{i=1}^{m} l_{ij} = 1 \cdot 1 = 1. \qquad \textbf{(2)}$$

(1) and (2) imply that $0 \le \tilde{x}(k) \le 1$ and the sum of entries in $\tilde{x}(k)$ equals 1. Therefore, sufficiency holds.

Necessity ($\Longleftarrow$)**:**

$$\sum_{i=1}^{m} \tilde{x}_i(k) = \sum_{j=1}^{n} x_j(k) \sum_{i=1}^{m} l_{ij} = \sum_{i=1}^{m} l_{ij}, j = 1, 2, \ldots, n \qquad \textbf{(3)}$$

If we want $\sum_i \tilde{x}_i(k) = 1$, then the right hand side of **(3)** is also equal to 1. $\sum_{i=1}^{m} l_{ij} = 1$ means every column in $\mathbb{L}$ has to sum to 1. Next we prove that every element in $\mathbb{L}$ has to be between $[0, 1]$. Suppose that not all the element in $\mathbb{L}$ is between [0,1]. Pick $l_{11} > 1$. Since $\mathbb{L}$ should satisfy any $x(0)$, we let $l_{11} > \frac{1}{x_1(k)}$ (similar proof also applies if $l_{11} < \frac{1}{x_1(k)}$).

On one hand,

$$\begin{aligned} & 0 \leq \tilde{x}_1(k) = l_{11}x_1(k) + l_{12}x_2(k) + \ldots + l_{1n}x_n(k) \leq 1 \\ \Longleftrightarrow\ & 0 \leq l_{11}x_1(k) \leq 1 - l_{12}x_2(k) - \ldots l_{1n}x_n(k) \leq 1 \qquad \textbf{(4)} \end{aligned}$$

One the other hand,

$$l_{11} > \tfrac{1}{x_1(k)} \Longleftrightarrow l_{11}x_1(k) > 1 \qquad \textbf{(5)}$$

(4) and **(5)** raise a contraction. Therefore, $0 \leq l_{ij} \leq 1$.

□

Example 2.7 This is an example showing that if $\mathbb{L}$ is not column stochastic (e.g. a non-weighted lumping operator), the lower order hidden state vector cannot be column stochastic. Given

$$A = \begin{bmatrix} 1/3 & 5/24 & 1/2 \\ 1/3 & 8/24 & 4/12 \\ 1/3 & 11/24 & 7/12 \end{bmatrix} \qquad C = \begin{bmatrix} 4/5 & 13/20 & 1/2 \\ 1/5 & 7/20 & 1/2 \end{bmatrix}$$

If we have a non-weighted operator

$$L_n = \begin{bmatrix} -3/4 & 9/16 & 15/8 \\ 7/4 & 7/16 & -7/8 \end{bmatrix}$$

we can obtain

$$\widetilde{A} = \begin{bmatrix} 1 & 3/4 \\ 0 & 1/4 \end{bmatrix} \qquad \widetilde{C} = \begin{bmatrix} 3/5 & 5/7 \\ 2/5 & 2/7 \end{bmatrix}.$$

Note that each column in L_n sums to 1, but L_n is not stochastic. This implies that there exists some $x(0)$ which makes $\tilde{x}(k)$ not stochastic. For instance, if choose $x(0) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T$, we get $\tilde{x}(0) = \begin{bmatrix} -1.75 & 0.75 \end{bmatrix}^T$. Though it seems that $\tilde{x}(k) = L_n x(k)$ would still hold and entries in $x(k)$ still sum to one, the negative elements prevent the entries in $\tilde{x}(k)$ being between $[0, 1]$.

□

Theorem 2.1.6 indicates that if we want to use lumping to obtain low-order realizations, weighted lumping is the furthest extent that we can go. Beyond that, stochasticity is not guaranteed.

2.1.4 Realizations with Weighted Lumping

In this section, we discuss how weighted lumping can help to determine the existence of low-order realizations given a true system and how transition and emission matrices of a low-order realization can be constructed under certain constraints with weighted lumping.

Lemma 2.1.7 Given a n^{th} order HMM (A, C) and a m^{th} order HMM $(\widetilde{A}, \widetilde{C})$, $m < n$,

$$x(k+1) = Ax(k), \qquad \widetilde{x}(k+1) = \widetilde{A}\widetilde{x}(k),$$
$$y(k) = Cx(k) \qquad \widetilde{y}(k) = \widetilde{C}\widetilde{x}(k)$$

if $\forall k, \forall x(0), \exists\ \mathbb{L}$ s.t. $\tilde{x}(k) = \mathbb{L}x(k)$, then $\widetilde{A}\mathbb{L} = \mathbb{L}A$.

Proof If the state vectors of two systems are related by $\mathbb{L}$, i.e.

$$\begin{aligned}
&\forall k, \forall x(0), \tilde{x}(k) = \mathbb{L}x(k) \\
\implies &\forall k, \forall x(0), \widetilde{A}^k \widetilde{x}(0) = \mathbb{L}A^k x(0) \\
\implies &\forall k, \forall x(0), \widetilde{A}^k \mathbb{L}x(0) = \mathbb{L}A^k x(0), \\
\implies &\forall k, \forall x(0), (\widetilde{A}^k \mathbb{L} - \mathbb{L}A^k)x(0) = 0, \\
\implies &\forall k, (\widetilde{A}^k \mathbb{L} - \mathbb{L}A^k) = 0, \\
\implies &k = 1, (\widetilde{A}\mathbb{L} - \mathbb{L}A) = 0 \\
\implies &\widetilde{A}\mathbb{L} = \mathbb{L}A
\end{aligned}$$

□

Lemma 2.1.8 Given a n^{th} order HMM (A, C) and a m^{th} order HMM $(\widetilde{A}$,$\widetilde{C})$, $m < n$,

$$\begin{aligned}
x(k+1) &= Ax(k), & \widetilde{x}(k+1) &= \widetilde{A}\widetilde{x}(k), \\
y(k) &= Cx(k) & \widetilde{y}(k) &= \widetilde{C}\widetilde{x}(k)
\end{aligned}$$

if $\forall k, \forall x(0)$, $\exists\ \mathbb{L}$ s.t. $\widetilde{A}\mathbb{L} = \mathbb{L}A$, then $\tilde{x}(k) = \mathbb{L}x(k)$ if and only if $\widetilde{A}[\tilde{x}(0) - \mathbb{L}x(0)] = 0$.

Proof If $\widetilde{A}\mathbb{L} = \mathbb{L}A$,$\forall k, \forall x(0)$, then

$$\begin{aligned}
\tilde{x}(k) - \mathbb{L}x(k) &= \widetilde{A}^k \widetilde{x}(0) - \mathbb{L}A^k x(0) \\
&= \widetilde{A}^k \widetilde{x}(0) - (\mathbb{L}A)A^{k-1}x(0) \\
&= \widetilde{A}^k \widetilde{x}(0) - \widetilde{A}\mathbb{L}A^{k-1}x(0) \\
&= \widetilde{A}^k \widetilde{x}(0) - \widetilde{A}(\mathbb{L}A)A^{k-2}x(0) \\
&= \widetilde{A}^k \widetilde{x}(0) - \widetilde{A}\widetilde{A}\mathbb{L}A^{k-2}x(0) \\
&= \dots \\
&= \widetilde{A}^k \widetilde{x}(0) - \widetilde{A}^k \mathbb{L}x(0) \\
&= \widetilde{A}^k [\widetilde{x}(0) - \mathbb{L}x(0)]
\end{aligned}$$

When $k = 1$, if $\widetilde{A}[\widetilde{x}(0) - \mathbb{L}x(0)] = 0$, then $\tilde{x}(k) - \mathbb{L}x(k) = 0$.

Note that if $\widetilde{A}$ has full rank, we only need to verify if $[\widetilde{x}(0) - \mathbb{L}x(0)] = 0$ to determine if $\tilde{x}(k) - \mathbb{L}x(k) = 0$.

Lemma 2.1.9 Given a n^{th} order HMM (A, C) and a m^{th} order HMM $(\widetilde{A}$,$\widetilde{C})$, $m < n$,

$$x(k+1) = Ax(k), \qquad \widetilde{x}(k+1) = \widetilde{A}\widetilde{x}(k)$$
$$y(k) = Cx(k), \qquad \widetilde{y}(k) = \widetilde{C}\widetilde{x}(k)$$

if $\forall k, \forall x(0)$, $\exists\, \mathbb{L}$ s.t. $\tilde{x}(k) = \mathbb{L}x(k)$, then $(\widetilde{A}, \widetilde{C})$ is a realization of (A, C) if $\widetilde{C}\mathbb{L} = C$.

Proof If $\widetilde{C}\mathbb{L} = C$, then $\tilde{y}(k) - y(k) = \widetilde{C}\tilde{x}(k) - Cx(k) = [\widetilde{C}\mathbb{L} - C]x(k) = 0, \forall k$. However, the converse is not true. This is because, if $(\widetilde{A}$,$\widetilde{C})$ is a realization of (A, C), then

$$\forall k, \forall x(0), \tilde{y}(k) - y(k) = 0,$$
$$\Longrightarrow \forall k, \forall x(0), [\widetilde{C}\mathbb{L} - C]x(k) = 0$$
$$\Longrightarrow \forall k, \forall x(0), [\widetilde{C}\mathbb{L} - C]A^k x(0) = 0$$
$$\Longrightarrow k = 1, \forall x(0), [\widetilde{C}\mathbb{L} - C]Ax(0) = 0$$

Since it holds for any $x(0)$, $[\widetilde{C}\mathbb{L} - C]A = 0$

Therefore, if $(\widetilde{A}, \widetilde{C})$ is a realization of (A, C) obtained from lumping, they have to satisfy $[\widetilde{C}\mathbb{L} - C]A = 0$.

□

Remark Particularly, if $\widetilde{C}L = C$ where L is an aggregation operator from integer lumping, then $rank(\widetilde{C}) = rank(C)$. To show this, we treat $\widetilde{C}L = C$ as a block of linear equations. On the left side, there are $2m$ variables in $\widetilde{C}$; on the right side, there are $2n$ equations. $m < n$ means the number of equations are more than the number of variables. Unless $rank(\widetilde{C}) = rank(C)$, there will be no solution for $\widetilde{C}L = C$.

The above lemmas provide criteria to determine if two HMMs of different orders are equivalent. In the following, we discuss what kind of emission probability matrix is able to generate a low-order realization by integer lumping.

Lemma 2.1.10 Given a n^{th} order HMM (A, C) and $L \in \mathbb{R}_+^{m\times n}$, where $C \in \mathbb{R}^{h\times n}$, $m < n$, if there is a low-order realization that can be found by integer lumping, then

C has to have m blocks, with each block having r_i columns the same as the i^{th} column in $\widetilde{C}$, where r_i is the length of $1s$ in the i^{th} row of L.

Proof From Lemma 2.1.2, we know L has full rank. Through linear row transformations, we can always rewrite L as:

$$L = \begin{bmatrix} 1_{11} & \dots & 1_{1r_1} & 0 & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & 0 \\ 0 & \dots & 0 & 1_{21} & \dots & 1_{2r_2} & 0 & \dots & \dots & \dots & \dots & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & \dots & \dots & \dots & \dots & 1_{i1} & \dots & 1_{ir_i} & 0 & \dots & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & 0 & 1_{m1} & \dots & 1_{mr_m} \end{bmatrix} \tag{2.5}$$

where the i^{th} row has a number of r_i $1s$ from 1_{i1} to 1_{ir_i}.

Let

$$\widetilde{C} = \begin{bmatrix} \widetilde{c}_{11} & \widetilde{c}_{12} & \dots & \widetilde{c}_{1m} \\ \widetilde{c}_{21} & \widetilde{c}_{22} & \dots & \widetilde{c}_{2m} \\ \dots & \dots & \dots & \dots \\ \widetilde{c}_{h1} & \widetilde{c}_{h2} & \dots & \widetilde{c}_{hm} \end{bmatrix}$$

denote the m^{th} low-order emission matrix at time instant k.

$$\widetilde{C}(k)L = \begin{bmatrix} \widetilde{c}_{11} & \dots & \widetilde{c}_{11} & \widetilde{c}_{12} & \dots & \widetilde{c}_{12} & \dots & \dots & \dots & \widetilde{c}_{1m} & \dots & \widetilde{c}_{1m} \\ \widetilde{c}_{21} & \dots & \widetilde{c}_{21} & \widetilde{c}_{22} & \dots & \widetilde{c}_{22} & \dots & \dots & \dots & \widetilde{c}_{2m} & \dots & \widetilde{c}_{2m} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \widetilde{c}_{h1} & \dots & \widetilde{c}_{h1} & \widetilde{c}_{h2} & \dots & \widetilde{c}_{h2} & \dots & \dots & \dots & \widetilde{c}_{hm} & \dots & \widetilde{c}_{hm} \end{bmatrix} = C \tag{2.6}$$

There are m blocks in (2.6). Within the i^{th} block, there are r_i identical columns. The number of $\widetilde{c}_{ij}$ in the i^{th} row of C is r_i as well. Thus we show that C has to take the form of equation (2.6) such that there is a solution to $\widetilde{C}L = C$.

□

Example 2.8 Given

$$L = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \text{ if } \widetilde{C}(k) = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix}(k).$$

Then

$$C = \begin{bmatrix} c_{11} & c_{11} & c_{12} \\ c_{21} & c_{21} & c_{22} \end{bmatrix}$$

□

In the above, we gave some criterion on determining the equivalence of two given HMMs assuming a lumping operator exists. A more practical question is: Given a HMM, how do we find a lumping operator and how do we use this operator to construct a low-order realization $(\widetilde{A}, \widetilde{C})$? In the following, we present a method to construct a low-order realization for an arbitrary true system with weighted lumping.

Theorem 2.1.11 Given a n^{th} order Markov chain $x(k+1) = Ax(k)$, $\exists\ L_w \in \mathbb{R}^{m\times n}$, s.t. $\exists \tilde{A} \in \mathbb{R}^{m\times m}$, $\tilde{x}(k+1) = \tilde{A}\tilde{x}(k)$ and $\tilde{x}(k) = L_w x(k)$.

Proof Construct

$$L_w = \begin{bmatrix} l_1 & l_1 & \dots & l_1 \\ l_2 & l_2 & \dots & l_2 \\ \dots & \dots & \dots & \dots \\ l_{m-1} & l_{m-1} & \dots & l_{m-1} \\ l_m & l_m & \dots & l_m \end{bmatrix}_{m\times n},$$

where $0 \leq l_i \leq 1$ and $\sum_{i=1}^{m} l_i = 1$.

Let $B = L_w A$ and denote b_{ij} as the entry in i^{th} row j^{th} column. Then we have

$$b_{ij} = l_i(a_{1j} + a_{2j} + \ldots + a_{nj}) = l_i \cdot 1 = l_i$$

We also calculate the j^{th} column sum of B:

$$\sum_{i=1}^{m} b_{ij} = \sum_{i=1}^{m} l_i = 1,\ j = 1, 2, \ldots, n$$

This indicates that each column in B is identical to each column in L_w. Thus, $B = L_w$, i.e., $L_w A = L_w$.

As a result, $L_w A^k = L_w A \cdot A^{k-1} = L_w A^{k-1} = \ldots = L_w$ **(6)**

Let

$$\tilde{A} = \begin{bmatrix} l_1 & l_1 & \ldots & l_1 \\ \ldots & \ldots & \ldots & \ldots \\ l_{m-1} & l_{m-1} & \ldots & l_{m-1} \\ l_m & l_m & \ldots & l_m \end{bmatrix}_{m \times m}$$

Similarly, we can prove that $\tilde{A}^k L_w = L_w$ **(7)**

Combining (6) and (7), $L_w A^k = \tilde{A}^k L_w = L_w$.

Therefore, $\tilde{A}^k L_w x(0) = L_w A^k x(0) \Rightarrow \tilde{x}(k) = L_w x(k)$.

□

In this way, we relax restrictions on A and build a system that can be any order lower than the true system, where its state vector can perform dynamically as the true system does. We show that weighted lumping extends lumping from a binary partition to a more general class associated with a probability framework. Instead of giving the interpretation of whether an entry belongs to a partition, we are now able to assign a probability to each entry in the true system to represent how likely each entry belongs to different clusters in the low-order system.

Example 2.9 Given

$$A = \begin{bmatrix} 0.42 & 0.42 & 0.31 & 0.41 \\ 0.10 & 0.36 & 0.17 & 0.33 \\ 0.27 & 0.21 & 0.23 & 0.08 \\ 0.21 & 0.01 & 0.29 & 0.18 \end{bmatrix} \text{ and } C = \begin{bmatrix} 0.5505 & 0.5505 & 0.5505 & 0.5505 \\ 0.4495 & 0.4495 & 0.4495 & 0.4495 \end{bmatrix},$$

using

$$L_w = \begin{bmatrix} 0.56 & 0.56 & 0.56 & 0.56 \\ 0.41 & 0.41 & 0.41 & 0.41 \\ 0.03 & 0.03 & 0.03 & 0.03 \end{bmatrix},$$

we can find

$$\widetilde{A} = \begin{bmatrix} 0.56 & 0.56 & 0.56 \\ 0.41 & 0.41 & 0.41 \\ 0.03 & 0.03 & 0.03 \end{bmatrix} \text{ and } \widetilde{C} = \begin{bmatrix} 0.58 & 0.50 & 0.69 \\ 0.42 & 0.50 & 0.31 \end{bmatrix}.$$

We verify that

$$L_w A = \widetilde{A} L_w = \begin{bmatrix} 0.56 & 0.56 & 0.56 & 0.56 \\ 0.41 & 0.41 & 0.41 & 0.41 \\ 0.03 & 0.03 & 0.03 & 0.03 \end{bmatrix} \text{ and } \widetilde{C} L_w = C.$$

□

Note that with this method, every column in the weighted lumping operator L_w is identical. Moreover, to ensure the existence of a low-order realization with weighted lumping, the emission matrix must have identical columns as well. If C does not have identical columns, it is unable to find a $\widetilde{C}$ such that $\widetilde{C} L_w = C$. However, a HMM satisfying this requirement of C will lose its dynamics. No matter how the hidden states change, the output $\widetilde{y}(k)$ will now take the values from one of the identical columns in C and stay in that distribution forever. In other words, there is no "hidden " information any more in the HMM.

It is also noted that the probabilities within each column of L_w can take any positive real numbers between $[0, 1]$. This implies that there are various weighted lumping operators which will lead to multiple low-order realizations.

In addition, if we examine the observability matrix O in the above example, rank(O)=1. From Lemma 2.1.1, we know that this true system is unobservable and non-minimal. Therefore, weighted lumping is simply an approach that can generate a low-order equivalent realization from non-minimal systems.

Chapter 3

Approximations of Minimal Systems

The previous chapter discussed equivalent realizations of two HMMs. We showed that generated by lumping or weighted lumping, systems of order lower than the true system but higher than the minimal system are actually equivalent realizations of unobservable HMMs. These unobservable HMMs are non-minimal systems. Compared to the realization problem, what is more interesting is to study approximations of minimal systems when they do not have low-order realizations.

3.1 Lower-order Approximations of Minimal Systems

An approximation problem is different from the realization problem in that there is not a low-order system than can generate the same output distribution as the true system. Rather than finding a realization that has a zero distance to the true system, the approximation problem aims at identifying low-order best approximations that have a minimal distance to the true system and characterizing the relationship between different low-order approximations of a minimal system. Furthermore, it is different from an estimation problem because instead of choosing a system given data as the learning algorithm does in estimating all the parameters, it selects an approximated system from the candidate systems given the true system.

Recall from the previous chapter the definition of an approximation: given a true system (A, C) of order n, $(\tilde{A}, \tilde{C})$ is said to be an *approximation* of (A, C) if the distance is nonzero; among all the systems that have nonzero distance to the true system (A, C), $(\tilde{A}, \tilde{C})$ is a *best approximation* of (A, C) if the distance from $(\tilde{A}, \tilde{C})$ to the true system is minimal. A best approximation problem can be written as:

$$\min_{(\tilde{A},\tilde{C})\in S} d((A, C), (\tilde{A}, \tilde{C}))$$

where S is a set of candidate systems of order $m < n$. Since we measure the distance between HMMs by their output distributions, the best approximation problem can be also written as:

$$\min_{(\tilde{A},\tilde{C})\in S} d((A, C), (\tilde{A}, \tilde{C})) = \min_{(\tilde{A},\tilde{C})\in S} \min_{\tilde{x_o}} \max_{x_o} \parallel y(k) - \tilde{y}(k) \parallel_2 \tag{3.1}$$

This criterion first assumes the worst case of an initial condition in the true system by using $\max\limits_{x_o}$. Another possible choice other than a *max* could be an average operation among outputs of all time steps. The reason why we choose a *max* operation is that the worst case of an initial condition choice of the true system would represent the potential upper bound of the distance between two systems. If we could find the minimal distance in the worst case, we might be able to put a constraint on the lower bound and determine the minimal distance for all the other cases.

Under the measurement of a 2-norm, the equation will search all the possible systems in the low-order candidate set and every possible low-order initial condition in the state space (using $\min\limits_{\tilde{x_o}}$) in order to minimize the difference between output distributions from the observation sequences.

The 2-norm of a vector $y = [y_1 \ldots y_n]^T$ is defined as:

$$\parallel y \parallel_2 = \sqrt{\sum_{i=1}^{n} |y_i|^2} \tag{3.2}$$

2-norm is one of the most commonly used norms and enjoys the geometric meaning of the sum of the total area under the curve of the vector. We could also use the infinity-norm, but we expect that the measuring results will be similar no matter what norm is chosen.

With this criterion, we are interested in describing the relationship among best approximations of different orders. As we discussed in the previous chapter that best approximations of order strictly less than that of a minimal system are truly approximations, they are unable to behave in an exact manner as the true system does. This means that the distance between two HMMs is non-zero. We want to explore how the distance from a best approximation to a true system varies as the order of an approximation changes. We prove in the following that the resulting approximation error is non-decreasing as the model order decreases, verifying our prediction that increasingly coarse models are less and less descriptive of the true system.

Theorem 3.1.1 Given a minimal system (A, C) of order n, and given two low-order systems $(\tilde{A}_{m_1}, \tilde{C}_{m_1})$ and $(\tilde{A}_{m_2}, \tilde{C}_{m_2})$, where $(\tilde{A}_{m_1}, \tilde{C}_{m_1})$ is a best approximation of (A, C) at order m_1, $(\tilde{A}_{m_2}, \tilde{C}_{m_2})$ is a best approximation of (A, C) at order m_2, $0 < m_2 < m_1 < n$, then $d((A, C), (\tilde{A}_{m_2}, \tilde{C}_{m_2})) \geq d((A, C), (\tilde{A}_{m_1}, \tilde{C}_{m_1}))$.

Proof Presume that the distance from $(\tilde{A}_{m_1}, \tilde{C}_{m_1})$ to the true system (A, C) is larger than the distance from $(\tilde{A}_{m_1}, \tilde{C}_{m_1})$ to (A, C), that is

$$d((A, C), (\tilde{A}_{m_2}, \tilde{C}_{m_2})) < d((A, C), (\tilde{A}_{m_1}, \tilde{C}_{m_1})) \tag{3.3}$$

$\forall \tilde{x}_{m_2}(0)$, construct another system $(\tilde{A}'_{m_1}, \tilde{C}'_{m_1})$ of order $m1$ as:

$$\tilde{x'}_{m_1}(k+1) = \tilde{A}'_{m_1}\tilde{x}'_{m_1}(k) = \begin{bmatrix} \tilde{A}_{m_2} & 0 \\ 0 & I_{m_1-m_2} \end{bmatrix} \tilde{x}'_{m_1}(k),$$

$$\tilde{y'}_{m_1}(k) = \tilde{C'}_{m_1}\tilde{x'}_{m_2}(k) = \begin{bmatrix} \tilde{C}_{m_2} & D_{2,m_1-m_2} \end{bmatrix} \tilde{x'}_{m_2}(k),$$

$$\tilde{x'}_{m_1}(0) = \begin{bmatrix} \tilde{x}_{m_2}(0) \\ 0 \end{bmatrix}$$

Then

$$\begin{aligned} \tilde{y'}_{m_1}(k) &= \tilde{C'}_{m_1}\tilde{A'}^k_{m_1}\tilde{x'}_{m_1}(0); \\ &= \begin{bmatrix} \tilde{C}_{m_2} & D_{2,m_1-m_2} \end{bmatrix} \begin{bmatrix} \tilde{A}_{m_2} & 0 \\ 0 & I_{m_1-m_2} \end{bmatrix}^k \begin{bmatrix} \tilde{x}_{m_2}(0) \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} \tilde{C}_{m_2}\tilde{A}^k_{m_2} & D_{2,n-m} \end{bmatrix} \begin{bmatrix} \tilde{x}_{m_2}(0) \\ 0 \end{bmatrix} \\ &= \tilde{C}_{m_2}\tilde{A}^k_{m_2}\tilde{x}_{m_2}(0) \\ &= \tilde{y}_{m_2}(k) \end{aligned}$$

This means $(\tilde{A'}_{m_1}, \tilde{C'}_{m_1})$ and $(\tilde{A}_{m_2}, \tilde{C}_{m_2})$ are equivalent realizations, that is

$$d((A, C), (\tilde{A}_{m2}, \tilde{C}_{m2})) = d((A, C), (\tilde{A'}_{m1}, \tilde{C'}_{m1})) \tag{3.4}$$

By the presumption in (3.3),

$$d((A, C), (\tilde{A}_{m_2}, \tilde{C}_{m_2})) < d((A, C), (\tilde{A}_{m_1}, \tilde{C}_{m_1}))$$

By substituting (3.4) into (3.3), we have

$$d((A, C), (\tilde{A'}_{m_1}, \tilde{C'}_{m_1})) < d((A, C), (\tilde{A}_{m_1}, \tilde{C}_{m_1})) \tag{3.5}$$

(3.5) suggests that other than $(\tilde{A}_{m_1}, \tilde{C}_{m_1})$, another system $(\tilde{A'}_{m_1}, \tilde{C'}_{m_1})$ can be found which has a shorter distance to the true system (A, C) than $(\tilde{A}_{m_2}, \tilde{C}_{m_2})$ does. It contradicts the definition of best approximation in which a best approximation of a minimal system always has the shortest distance to the true system. Therefore,

$$d((A, C), (\tilde{A}_{m_2}, \tilde{C}_{m_2})) \geq d((A, C), (\tilde{A}_{m_1}, \tilde{C}_{m_1})).$$

In other words, as the order of approximations becomes lower, the distance between a best approximation and a true system is nondecreasing. It also indicates that the

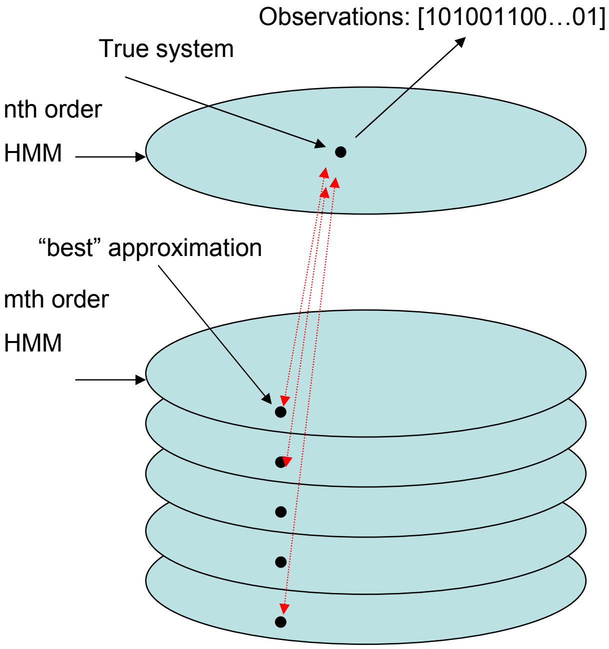


Figure 3.1: **Approximation errors of different order systems**

distance from one best approximation to another best approximation of a different order is nonnegative (see Figure 3.1).

□

Lemma 3.1.1 indicates that, more and more information of the true system is losing when the size of an approximation is shrinking; systems with decreasing orders are less and less descriptive of the true system.

Chapter 4

Conclusions and Future Work

This thesis characterizes the quality of Hidden Markov modeling when learning from limited data. It introduces a new perspective to describe the sources of under-modeling errors by decomposing the modeling errors into two fundamental categories: first, the approximation error which describes the distance between the true system and a system of order lower than the true system; and second, the estimation error which refers to the modeling error coming from the learning algorithm with finite observational data. This thesis investigates the approximation error of low order HMMs and further classifies a best approximation into an approximation and a realization.

4.1 Conclusions

Best approximations which assume the smallest distance from the approximated system to the true system can be described by the notion of minimality; realizations, as a special type of approximations with complexity greater or equal to the order of a minimal system, are equivalent to the true system. The equivalence of two systems is evaluated by the notion of dynamic consistency and a distance measurement is provided to calculate the minimality. To show the relationship between minimality and observability, this thesis proves that an observable system is a minimal system. For realizations of non-minimal systems, this thesis also examines the properties of integer lumping and presents a more general method named weighted lumping to construct realizations of a non-minimal system. Examples and experiments have been conducted with this method and results show that a stochastic transition matrix can be randomly chosen and turned into a system of lower order such that hidden states between the true system and the low order system are dynamically related by a particular aggregation operator. It is also shown that best approximations of order strictly less than that of a minimal realization are truly approximations; they are usually unable to precisely reproduce the output distribution of the true system. The work then proves that the resulting approximation error is non-decreasing as the model order decreases, which verifies the intuitive idea that increasingly simplified models are less and less descriptive of the true system.

4.2 Future Work

As a next step to characterize HMM undermodeling errors, we would like to prove the strict monotonicity of distance from the best approximation to the true system as the order of the approximation decreases. We have already proved the non-decreasing property of the distance. A future proof of an increasing distance property would be very important because it would help to more precisely describe the tradeoff between uncertainty and complexity. We would also like to explicitly define the mathematical meaning of precision and accuracy in order to refine the entire modeling error picture in terms of complexity. Furthermore, we would like to examine order estimation approaches more closely and investigate the possible existence of a unique minimal-error system which would be able to mimic the true system to the greatest extent with limited data. One of our ultimate goals is to verify the entire picture we proposed as it is shown in Figure 4.1. We would like to show that not only the the approximation error is increasing, but also the estimation error is decreasing as the order of the system decreases; with these two modeling errors, we would be able to describe the

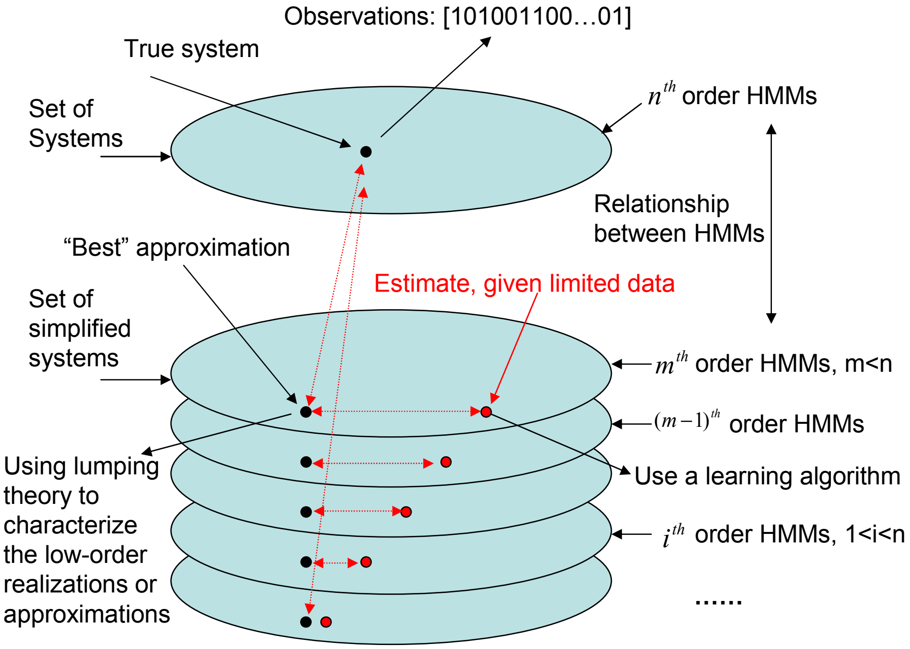

Observations: [101001100…01]
True system
Set of Systems
n^{th} order HMMs
Relationship between HMMs
"Best" approximation
Estimate, given limited data
Set of simplified systems
m^{th} order HMMs, m<n
$(m-1)^{th}$ order HMMs
Use a learning algorithm
Using lumping theory to characterize the low-order realizations or approximations
i^{th} order HMMs, 1<i<n
……

Figure 4.1: **The whole picture of three types of modeling errors**

modeling errors between the true system and the system estimated from the learning algorithm.

In terms of weighted lumping, we want to find a more general description of emission matrices such that the restricted scope of HMM choices can be released. In addition, we are looking for various applications of our approximation model in the fields of ecology, bioinformatics, and hydrology so that we will be able to examine its practicability in simplifying complicated natural systems and phenomena into abstracted computable mathematical models.

Chapter 5

Appendix

5.1 Transform a n^{th} order difference equation to n first-order equations

Proof An n^{th} order difference equation

$$y(k+1) = a_1 y(k) + a_2 y(k-1) + \ldots + a_n y(k-n+1)$$

can be written as a system of n first-order difference equations by defining vector

$$z(k+1) = \begin{bmatrix} y(k+1) \\ y(k) \\ \ldots \\ y(k-n+2) \end{bmatrix}$$

Rewrite $z(k+1)$ as

$$z(k+1) = \begin{bmatrix} y(k+1) \\ y(k) \\ \ldots \\ y(k-n+2) \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & \ldots & \ldots & a_n \\ 1 & 0 & \ldots & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & 1 & 0 \end{bmatrix} \begin{bmatrix} y(k) \\ y(k-1) \\ \ldots \\ y(k-n+1) \end{bmatrix} = Az(k)$$

Thus, the original n^{th} order system $y(k) = \sum_{i=1}^{n} a_i y(k-i+1)$ can now be represented as $z(k+1) = Az(k)$, which is a first-order system.

□

5.2 Derivation of the second condition in general lumping

$$\tilde{a}_{pq}(k) = \sum_{j:s_j \in \bar{S}_q} \lambda_{jq}(k) \sum_{i:s_i \in \bar{S}_p} a_{ij}, \text{ where } \lambda_{jq}(k) = Pr[s_j | s_h \in \bar{S}_q]$$

Proof .

Let s_i denote the node representing the i^{th} state in the state vector $x(k)$ at time k and s_i^{k+1} denotes the i^{th} node at time $(k+1)$. For node s_i grouped in cluster $\bar{S}_p$ and node s_j in cluster $\bar{S}_q$, according to the Total Probability Theorem[50], the transition probability $\tilde{a}_{pq}(k)$ from cluster $\bar{S}_q$ to $\bar{S}_p$ in the low-order model at time k is:

$$\begin{aligned} \tilde{a}_{pq}(k) &= Pr[s_i^{k+1} \in \bar{S}_p | s_h \in \bar{S}_q] \\ &= \sum_{j:s_j \in \bar{S}_q} Pr[s_i^{k+1} \in \bar{S}_p | s_j] \cdot Pr[s_j | s_h \in \bar{S}_q] \end{aligned}$$

This transition probability can be thought of as: summing up all the possible ways that every current node s_j in cluster $\bar{S}_q$ transfers to an arbitrary node in cluster $\bar{S}_p$. This probability is also the conditional probability that s_j transfers to any other node in cluster $\bar{S}_p$ given the current state node s_j belongs to cluster $\bar{S}_q$. Let $\lambda_{jq}(k)$ denote the conditional probability of node s_j in cluster $\bar{S}_q$ at time k,

$$\lambda_{jq}(k) = Pr[s_j | s_h \in \bar{S}_q] = \frac{Pr[s_h \in \bar{S}_q | s_j] \cdot Pr[s_j]}{Pr[s_h \in \bar{S}_q]} \tag{5.1}$$

For a specified partition, the probability that whether the current state node belongs to a cluster or not is either 1 or 0, i.e. $Pr[s_h \in \bar{S}_q | s_j] = 1$ if $s_j \in \bar{S}_q$ and $Pr[s_h \in \bar{S}_q | s_j] = 0$ if $s_j \not\subseteq \bar{S}_q$. Thus the nonzero part of λ should be:

$$\lambda_{jq}(k) = Pr[s_j | s_h \in \bar{S}_q] = \frac{Pr[s_j]}{Pr[s_h \in \bar{S}_q]} = \frac{x_j(k)}{\tilde{x}_m(k)} \tag{5.2}$$

where $x_j(k)$ is the j^{th} hidden state value before aggregation and $\tilde{x}_q(k)$ is the q^{th} state value after aggregation.

From the definition of Markov chains,

$$Pr[s_i^{k+1} \in \bar{S}_p | s_j] = \sum_{i:s_i \in \bar{S}_p} Pr[s_i^{k+1} | s_j] = \sum_{i:s_i \in \bar{S}_p} a_{ij} \tag{5.3}$$

Therefore, $\tilde{a}_{pq}(k) = \sum_{j:s_j \in \bar{S}_q} \lambda_{jq}(k) \sum_{i:s_i \in \bar{S}_p} a_{ij}$, where $\lambda_{jq}(k) = Pr[s_j | s_h \in \bar{S}_q]$.

□

When a transition matrix is partitioned into sub-matrices, this formula first sums up the entries column-wisely within each sub-matrix in the partition. Then by multiplying the sum with the corresponding conditional probability, the aggregated transition probability can be calculated. Conditional probabilities serve as weights on the atomic entries in the original system.

LIST OF REFERENCES

[1] L.R.Bahl, F. Jelinek, and R.L.Mercer, “A maximum likelihood approach to continuous speech recognition,” *IEEE Trans. Pattern Anal. Machine Intel.*, vol. PAMI-5, pp. 179–190, March 1983.

[2] F.Jelinek, *Statistical Methods for Speech Recognition.* Cambridge, MA: MIT Press, 1998.

[3] D.M.Goblirsch and N.Farvardin, “Switched scalar quantizers for Hidden Markov sources,” *IEEE Trans. Inform. Theory*, vol. 38, pp. 1455–1473, September 1992.

[4] I.L.MacDonald and W. Zucchini, *Hidden Markov and Other Models for Discrete-Valued Time Series.* London,U.K.:Chapman and Hall, 1997.

[5] A. e. a. Krogh, “Hidden Markov Models in computational biology: applications to protein modelling,” *Journal of Molecular Biology*, vol. 235, pp. 1501–1531, 1994.

[6] Q. Sun, D. R. Simon, Y.-M. Wang, W. Russell, V. N. Padmanabhan, and L. Qiu, “Statistical identification of encrypted web browsing traffic,” *Proceedings of the 2002 IEEE Symposium on Security and Privacy*, p. 19, May 12-15, 2002.

[7] L. Rabiner and B.H.Juang, *Fundamentals of Speech Recognition.* Prentice Hall, 1993.

[8] J. Wootton, “Prediction in complex communities: analysis of empirically-derived Markov models,” *Ecology*, vol. 82, pp. 580–598, 2001.

[9] Y. Ephraim and N. Merhav, “Hidden Markov processes,” *IEEE Transactions on Information Theory*, vol. 48, no. 6, pp. 1518–1569, June 2002.

[10] J.R.Norris, *Markov Chains.* Cambridge University Press, 1997.

[11] W. S. Levine, *The Control Handbook.* IEEE Press, 1995.

[12] L.E.Baum and T. Petrie, “Statistical inference for probabilitic functions of finite state Markov chains,” *Ann. Math. Statist.*, vol. 37, pp. 1554–1563, 1966.

[13] T.Petrie, “Probabilitistic functions of finite state Markov chains,” *Ann. Math. Statist.*, vol. 40, no. 1, pp. 97–115, 1969.

[14] L. Xu and M. I. Jordan, “On convergence properties of the EM algorithm for Gaussian Mixtures,” *Neural Computation*, vol. 8, no. 1, pp. 129–151, 1996. [Online]. Available: citeseer.ist.psu.edu/xu95convergence.html

[15] H. Ito, S.-I. Amari, and K. Kobayashi, “Identifiability of Hidden Markov information sources and their minimum degrees of freedom,” *IEEE Transactions on Information Theory*, vol. 38, no. 2, pp. 324–333, March 1992.

[16] A. G. R.Durbin, S. Eddy, *Biological sequence analysis: Probabilistic models of proteins and nucleic acids.* Cambridge University Press, 1998.

[17] R.J.Elliott, L.Aggoun, and J. Moore, *Hidden Markov Models: Estimation and Control.* New York: Springer-Verlag, 1994.

[18] R. A. Horn and C. R. Johson, *Matrix Analysis.* Cambridge University Press, 1985.

[19] D. G. Luenberger, *Introduction to Dynamic Systems: Theory, Models and Applications.* John Wiley and Sons, Inc., 1979.

[20] M. Zhou and J. Buongiorno, "Forestr landscape management in a stochastic environment, with an application to mixed loblolly pine-hardwood forests," *Forest Ecology and Management*, vol. 223, pp. 170–182, October 2005.

[21] L.Finesso, "Consistent estimation of the order for markov and hidden markov chains," Ph.D. dissertation, Univ. Maryland, College Park, 1990.

[22] J.C.Kieffer, "Strongly consistent code-based identification and order estimation for constrained finite-state model classes," *IEEE Trans. Inform. Theory*, vol. 39, pp. 893–902, May 1993.

[23] J.Ziv and N.Merhav, "Estimating the number of states of a finite-state souce," *IEEE Trans. Inform. Theory*, vol. 38, pp. 61–65, January 1992.

[24] D.Blackwell and L.Koopmans, "On the identifiability problem for functions of finite Markov chains," *Annals of Mathematical Statistics*, vol. 28, pp. 1011–1015, 1957.

[25] G.Picci, "Recent developments in variable structure systems," *Lecture Notes in Economic and Mathematical Systems*, vol. 162, pp. 288–304, 1978.

[26] J. van den Hof and J. van Schuppen, "Realization of positive linear systems using polyhedral cones," *Proceedings of the 33rd Conference on Decision and Control*, vol. FP-8 4:30, pp. 3889–3893, December 1994.

[27] E.J.Gilbert, "On the identifiability problem for functions of finite Markov chains," *Ann. Math. Statist.*, vol. 30, pp. 688–697, 1959.

[28] R.E.Kalman and M.A.Arbib, *Topics in Mathematical System Theory.* New York: McGraw-Hill, 1969.

[29] M.Fox, “Conditions under which a given process is a functions of a Markov chain,” *Ann. Math. Statist.*, vol. 33, p. 1206, 1962.

[30] S.W.Dharmadhikari and M.G.Nadkarni, “Some regular and nonregular functions of finite Markov chains,” *Ann. Math. Statist.*, vol. 41, pp. 207–213, 1970.

[31] H.A.Simon and A.Ando, “Aggregation of variables in dynaical systems,” *Econometrica*, vol. 28, pp. 111–138, 1961.

[32] K. Dogancay and V. Krishnamurthy, “Quick aggregation of markov chain functionals via stochastic complementation,” *Proc. IEEE Int. Conf. Acoustics, Speech Signal Processing*, vol. 1, pp. 63–66, 1997.

[33] Y. Pribadi, J. Voeten, and B. Theelen, “Reducing Markov chains for performance evaluation,” *Proceedings of PROGRESS’01. Utrecht (The Netherlands):STW Technology Foundation*, pp. 173–179, 2001. [Online]. Available: citeseer.ist.psu.edu/pribadi01reducing.html

[34] H. C.Tijms, *Stochastic Models: An Algorithmic Approach.* John Wiley and Sons, 1994.

[35] J. G. Kemeny and J. L. Snell, *Finite Markov Chains.* Princeton, NJ, Van Nostrand, 1960.

[36] G. Kotsalis and M. Dahleh, “Model reduction of irreducible Markov chains,” *Proceedings of the 42nd IEEE Conference on Decision and Control*, vol. 6, pp. 5727–5728, December 2003.

[37] L.B.White, R.Mahon, and G.D.Brushe, “Lumpable Hidden Markov Models-model reduction and reduced complexity filtering,” *IEEE Transactions on Automatic Control*, vol. 45, no. 12, 2000.

[38] A.Sokolova and E. de Vink, "On relational properties of lumpability," *Proceedings of the 4th PROGRESS Symposiu on Embedded Systems (Niewegein, STW Tehnology Foundation, The Netherlands)*, October 22, 2003.

[39] M. A. D. Soosan Beheshti, "A new information theoretic approach to order estimation problem," *13th IFAC Symposium on System Identification*, August 2003.

[40] J. M. van den Hof, "Realization of positive linear systems," *Linear Algebra and its Applications*, vol. 93B15, 15A48, 1991.

[41] ——, "Positive linear observers for linear compartmental systems," *SIAM J. Control Optim.*, vol. 36, pp. 590–608, 1998.

[42] L.Benvenuti, L.Farina, B.D.O.Anderson, and F. Bruyne, "Minimal positive realizations of transfer functions with positive real poles," *IEEE Trans. Circuits Syst. I*, vol. 47, pp. 1370–1377, September 2000.

[43] B.Nagy and M.Matolcsi, "A lowerbound on the dimension of positvie realizations," *IEEE Trans. Circuits Syst. I*, vol. 50, pp. 782–784, 2003.

[44] "The realization problem for Hidden Markov Models," *Math. Control, Signals, Syst.*, vol. 12, pp. 80–120, 1999.

[45] "Nonnegative realizations of matrix transfer functions," *Linear Alg. Applicat.*, vol. 311, pp. 107–129, 2000.

[46] L.Benvenuti and L.Farina, "A tutorial on the positive realization problem," *IEEE Transactions on automatic control*, vol. 49, no. 5, pp. 651–664, May 2004.

[47] Y. Kamp, "State reduction in hidden Markov chains used for speech recognition," *IEEE Transcations on Acoustics, Speech, and Signal Processing*, vol. 33, no. 5, pp. 1138–1145, October 1985.

[48] G.D.Brushe and L.B.White, “Spatial filtering of superimposed convolutional coded signals,” *IEEE Trans. Commun.*, vol. 45, pp. 1144–1153, September 1997.

[49] J.Ledoux, “On weak lumpability of denumerable Markov chains,” *Statistics and Probability Letters*, vol. 25, pp. 329–339, 1995.

[50] L. J. Bain and M. Engelhardt, *Introduction to Probability and Mathematical Statistics.* Duxbury, 1991.

[51] S. R. Eddy, “Profile Hidden Markov Models (review).” *Bioinformatics*, vol. 14, no. 9, pp. 755–763, 1998. [Online]. Available: citeseer.ist.psu.edu/eddy98profile.html

[52] P. Baldi and S. Brunak, *Bioinformatics: The Machine Learning Approach.* MIT Press, Cambridge, Massachusetts, 1998.

[53] W. R. Ingvar Eidhammer, Inge Jonassen, *Protein Bioinformatics: An Algorithmic Approach to Sequence and Structure Analysis.* John Wiley and Sons, Ltd, 2004.

[54] T. A. L. Bradley P. Carlin, *Bayes and Empirical Bayes Methods for Data Analysis.* Chapman and Hall, 1996.

[55] L. Ljung, *System Identification: Theory for the User.* Prentice Hall, 1987.

[56] S. Lynch, *Dynamical Systems with Applications using MATLAB.* Birkhauser, 2003.

[57] L.E.Baum, “An inequality and associated maximization technique in statistical estimation for probability functions of Makov processes.” *Inequalities*, vol. 3, pp. 1–8, 1972.

[58] T. M. Mitchell, *Machine Learning.* McGraw-Hill, 1997.

[59] T. K. Moon and W. C. Stirling, *Mathematical Methods and Algorithms for Signal Processing.* Prentice Hall, 2000.

[60] Little, R.J.A., and D.B.Rubin, “On jointly estimating parameters and missing data by maximizing the complete-data likelihood,” *Am. Statistn.*, vol. 37(3), pp. 218–220, 1983.

[61] Dempster, A. N.M.Laird, and D.B.Rubin, “Maximum Likelihood from incomplete data via the EM algorithm,” *J.Royal Statistical Soc.,Ser.B*, vol. 39(1), pp. 1–38, 1977.

[62] Schmee, J., and G.J.Hahn, “Simple method for regression analysis with censored data,” *Technometrics*, vol. 21(4), pp. 417–32, 1979.

[63] D. Meng, X.-L.and Rubin, “Recent extensions to the em algorithm,” *Bayesian Statistics*, vol. 4, pp. 307–320, 1992.

[64] M. I. Aleksandar Kavcic and F. I. Jose M.F.Moura, “The viterbi algorithm and markov noise memory,” *IEEE Transactions on Information Theory*, vol. 46, no. 1, pp. 291–301, January 2000.

[65] D. L. Weakliem, “A critique of the bayesian information criterion for model selection,” *Sociological Methods and Research*, vol. 27, no. 3, pp. 359–397, 1999.

[66] G.Schwarz, “Estimating the dimention of a model,” *The Annals of Statistics*, vol. 6, pp. 461–464, 1978.

[67] A. H., “A new look at the statistical model identification,” *IEEE Trans. Automat. Contr.*, vol. 19, pp. 716–23, 1974.

[68] Barron, A, Rissanen, J, Yu, and B, “The minimum description length principle in coding and modeling,” *Information Theory: 50 Years of Discovery(invited paper)*, vol. 44, no. 6, pp. 2094–2123, October 1998.

[69] D. L. Urban, “Modeling ecological processes across scales,” *Ecology*, vol. 86, no. 8, pp. 1996–2006, 2004.

[70] M. Rosenblatt, *Markov Process: Structure and Asymptotic Behavior.* Berlin: Springer-Verlag, 1971.

[71] B. D. O. Anderson, “The realization problem for Hidden Markov Models,” *Mathematics of Control, Signals and Systems*, vol. 12, pp. 80–120, 1999.

[72] S. Dharmadhikari, “Sufficient conditions for a stationary process to be a function of a finite Markov chain,” *Annals of Mathematical Statistics*, vol. 34, pp. 1033–1041, 1963.

[73] ——, “A characterization of a class of functions of finite Markov chains,” *Annals of Mathematical Statistics*, vol. 36, pp. 524–528, 1965.